UNIVERSIDADE DE BRASÍLIA – UNB

INSTITUTO DE LETRAS - IL

DEPARTAMENTO DE LÍNGUAS ESTRANGEIRAS E TRADUÇÃO - LET

LÍNGUAS ESTRANGEIRAS APLICADAS AO MULTILINGUISMO E À SOCIEDADE DA INFORMAÇÃO - LEA-MSI

MÔNICA CARDOSO SILVA

A INSERÇÃO DO ASURINÍ DO TROCARÁ NO MEIO DIGITAL:

PROPOSTAS EM PROL DA REVITALIZAÇÃO/FORTALECIMENTO DE LÍNGUAS INDÍGENAS

Brasília - DF

2025

MÔNICA CARDOSO SILVA

A INSERÇÃO DO ASURINÍ DO TROCARÁ NO MEIO DIGITAL:

PROPOSTAS EM PROL DA REVITALIZAÇÃO/FORTALECIMENTO DE LÍNGUAS INDÍGENAS

Trabalho de conclusão de curso apresentado ao Departamento de Línguas Estrangeiras e Tradução da Universidade de Brasília, como requisito para a obtenção do título de bacharel em Línguas Estrangeiras Aplicadas ao Multilinguismo e à Sociedade da Informação.

Orientadora: Clarissa Prado Marini

BRASÍLIA-DF

2025

A INSERÇÃO DO ASURINÍ DO TROCARÁ NO MEIO DIGITAL:

PROPOSTAS EM PROL DA REVITALIZAÇÃO/FORTALECIMENTO DE LÍNGUAS INDÍGENAS

Mônica Cardoso Silva

RESUMO
O trabalho visa introduzir a língua indígena dos Asurinís do Trocará, também conhecidos como Asurinís do Tocantins, e propor métodos que podem auxiliar na tarefa de revitalização da língua. Dentre eles, está a criação de um dicionário digital, um tradutor automático e um site próprio para o Asuriní, cujos códigos em linguagem de programação e recursos utilizados serão devidamente descritos conforme as ferramentas forem listadas. Os métodos apresentados são centrados no ciberespaço — isto é, o ambiente digital — por se tratar de um meio acessível e abrangente, com diversas possibilidades de revitalização linguística. Tendo em vista a pequena quantidade de falantes nativos de Asuriní, somado ao fato de que o mesmo está sendo cada vez menos ensinado às gerações atuais, é importante que iniciativas de apoio sejam feitas, para evitar com que desapareça. Sendo assim, essa pesquisa foi realizada para prestar apoio a quaisquer pessoas/profissionais interessados na causa, e que queiram colaborar para a revitalização ou estudo do idioma.
Palavras-chave: Asuriní; Línguas Indígenas; Revitalização; Ciberespaço.

ABSTRACT
The purpose of this paper is to introduce the indigenous language of the Asurinís of Trocará, also known as the Asurinís of Tocantins, and to propose methods that can assist in the task of language revitalisation. The options include the creation of a digital dictionary, a translator and a website dedicated to the Asuriní, with the programming language codes and resources applied being duly described as the proposals are listed. The tools and resources presented are centred on cyberspace, that is, the digital space, as it is an accessible and broad environment, with many possibilities for language revitalisation. Given the small number of native speakers of Asuriní, coupled with the fact that it is no longer being taught to current generations as before, it is important that support initiatives are taken to prevent it from disappearing. Therefore, this research was carried out to provide support to any people/professionals interested in the matter, whether to help in the revitalisation or simply studying the language.
Keywords: Asuriní; Indegenous Languages; Revitalisation; Cyberspace.

RÉSUMÉ
L'objectif de cet exposé est de présenter la langue autochtone des Asurinís du Trocará, également connus sous le nom d'Asurinís du Tocantins, et de proposer des méthodologies qui peuvent aider à la revitalisation de la langue. Les options comprennent la création d'un dictionnaire digital, d'un traducteur automatique et d'un site web pour l'Asuriní, dont les codes de langage de programmation et les ressources seront décrits au fur et à mesure que les propositions seront énoncées. Les outils et ressources présentés sont centrés sur le cyberespace, c'est-à-dire l'espace digital, car il s'agit d'un environnement accessible et vaste, offrant de nombreuses possibilités pour la revitalisation des langues. En raison du faible nombre de locuteurs natifs de l'Asuriní et du fait que cette langue n'est plus enseignée aux générations actuelles comme auparavant, il est important que des initiatives de soutien soient prises pour éviter qu'elle ne disparaisse. Ainsi, cette recherche a été menée dans le but d'apporter un soutien aux personnes/professionnels intéressés par ce thème, que ce soit pour aider à la revitalisation ou simplement à l'étude de la langue.
Mots-clés : Asuriní ; Langues Autochtones ; Revitalisation ; Cyberespace.

1. INTRODUÇÃO

No Brasil, existem inúmeras línguas indígenas espalhadas por todo o território brasileiro. Apesar de representarem uma pequena parcela de falantes, se comparadas com o português, ainda assim compõem uma imensa diversidade linguística. De acordo com uma pesquisa feita em 2021 pela Organização de Estados Ibero-Americanos, o Brasil possui uma das maiores diversidades

linguísticas de povos originários da América Latina, entre 110 e 190 línguas, isso quando não é considerado o país detentor da maior parte delas (variando de acordo com o senso demolinguístico). Apesar disso, apenas uma pequena parcela de nativos é falante de alguma língua indígena.

O censo do IBGE de 2022 revela que a média populacional de povos originários em território brasileiro é de cerca de um milhão e meio de pessoas, o que corresponde a apenas 0,83% da população total, sendo a região norte a responsável por abrigar a maior densidade demográfica do país. Esses dados revelam, portanto, que apesar da diversidade linguística, os idiomas nativos que ainda são falados estão concentrados em uma pequena parcela da população, o que colocaria diversas línguas indígenas na categoria de línguas minoritárias, ou seja: línguas que abarcam uma pequena parcela da população que possui uma língua diferente da considerada oficial, isto é, a língua majoritária (LIMBERGER; KÜRSCHNER; ALTENHOFEN; MOZZILLO, 2020).

Dados como esse, portanto, suscitam a ideia de que esses povos, a cultura e até mesmo a língua podem desaparecer com o tempo, tendo em vista que são grupos constantemente ameaçados. Diante do exposto, uma forma de prevenir o desparecimento, no que tange à língua, é fazer com que ela seja submetida a um processo de revitalização, ou seja, medidas capazes de reverter o declínio no número de falantes, seja por documentação linguística ou até mesmo com incentivo social para que o idioma continue sendo utilizado.

Uma dessas línguas ameaçadas é o Asuriní do Trocará, ou Asuriní do Tocantins, uma língua indígena falada por povos originários da região norte do país, no Estado do Pará. Sua condição de ameaçada se deve à quantidade ínfima de falantes que a utiliza como primeira língua, onde apenas algumas centenas dentre os aproximados 500 habitantes da aldeia a utilizam (CABRAL; SILVA; MARTINS; LOPES; CARVALHO; SOUSA, 2011).

Dada a emergência da situação, esse estudo foi realizado, com o intuito de prestar apoio ao idioma. Serão abordados métodos que podem auxiliar na preservação da língua, especificamente no que tange o meio digital, também conhecido como ciberespaço. A vantagem de utilizar desse meio para a tarefa de revitalização está diretamente relacionada com a ampla diversidade de tecnologias e fácil acesso que o computador e a própria internet têm a oferecer, podendo ser acessado em qualquer lugar e por quaisquer pessoas que também estejam interessadas na causa.

Um exemplo de idioma que se beneficiou do suporte online foi o havaiano, com a utilização do *Bulletin Board System*, que é um software de interação online (semelhante à internet), permitindo que a língua fosse trabalhada de forma digital. Além disso, algumas iniciativas foram feitas para documentar o havaiano, como o upload de materiais relativos à cultura, à língua e até mesmo sobre a comunidade, ademais de gravações de áudio e filmagens realizadas com os nativos se comunicando, com o intuito de deixar documentado o registro oral (WARSCHAUER, 1998).

Dentre as possibilidades a serem trabalhadas no ciberespaço, a pesquisa se concentrará na utilização de linguagem de programação, o uso de tradutores automáticos, a utilização de corpora, sites, inteligência artificial e dicionários. Além disso, os exemplos propostos ao longo da pesquisa advêm do material bibliográfico, não sendo criados termos e palavras provindas de outras fontes.

Além disso, esse trabalho é um desmembramento do Projeto de Iniciação Científica realizado em 2023 sobre a Implementação de um Etiquetador Automático para Nomes na Língua Asuriní do Tocantins. A contribuição do Laboratório de Línguas e Literatura Indígenas (LALLI) da Universidade de Brasília foi essencial para que esses estudos fossem realizados, tendo em vista que parte dos materiais bibliográficos utilizados se restringem ao uso interno do laboratório, além da permissão para poder trabalhar com o idioma.

2. A COMUNIDADE E A LÍNGUA ASURINÍ

A aldeia do Trocará está situada às margens do Rio Tocantins, no estado do Pará. Segundo dados do programa Povos Indígenas no Brasil, um portal vinculado ao Instituto Socioambiental, os Asurinís costumavam habitar a região do Xingu, esta que, por sua vez, abriga a aldeia dos Asurinís do Xingu, pertencentes à Terra Indígena Koatinemo. Apesar de ambas falarem Asuriní, existem diferenças morfológicas, fonológicas e gramaticais entre elas (POVOS INDÍGENAS NO BRASIL, 2024).

O tronco linguístico do Asuriní é o Tupí-Guaraní, sendo uma das mais conservadoras do ramo, principalmente no que se refere à gramática (CABRAL; SILVA; MARTINS; LOPES; CARVALHO; SOUSA, 2011). Ela é considerada uma língua não-configuracional, isto é, não há como determinar a função sintática de uma palavra pela posição que ela ocupa em uma oração. Para exemplificar, a língua portuguesa possui o seguinte padrão: substantivo, verbo e objeto

(S.V.O); logo, é possível inferir que o primeiro termo de uma oração seja o sujeito. O mesmo, no entanto, não se aplica ao Asuriní, já que o sujeito pode vir em qualquer posição numa frase.

Portanto, para identificar a função sintática que uma determinada palavra desempenha numa oração, é importante analisar os morfemas que a acompanham, isto é, os afixos. Eles podem ser definidos como partículas que se ligam aos radicais de uma palavra para alterar o sentido. Para exemplificar, observe as seguintes frases em Asuriní retiradas do dicionário Asuriní-Português (CABRAL; RODRIGUES, 2003):

okáj ka'á - o mato queimou

O substantivo "ka'á" - mato, apareceu no fim da frase, e ele desempenha a função sintática de sujeito da oração. Esse caso, no entanto, não ocorre na seguinte frase:

na sé resángihi ka'ápe - ele não me viu no mato

Nesse caso, a palavra "mato" também está localizada no fim da oração, porém, desta vez, acompanhada pelo locativo pontual "-pe", atribuindo o sentido de adjunto adverbial de lugar "no mato". Porém, a mesma frase também pode ser reescrita em uma ordem diferente, como o exemplo a seguir mostra:

ka'ápe na seresángihi - no mato ele não me viu

Apesar de exercerem diferentes funções sintáticas, a palavra continua sendo um substantivo, ou seja, não há mudança morfológica nesse caso. Entretanto, também é possível alterar a classe gramatical de uma palavra com a utilização de afixos, como é o caso do verbo "soká", "matar", que pode se transformar no substantivo "o morto" ao adicionar o sufixo "-pýt".

Essa análise gramatical é importante porque uma das formas de se trabalhar com línguas no ciberespaço é utilizando a linguagem de programação, esta que, por sua vez, necessita reconhecer as nuances da língua para então classificar corretamente as palavras e estruturas

gramaticais. Entretanto, antes de adentrar no tópico da revitalização em si, é importante salientar o porquê do idioma necessitar de tais meios de preservação.

Além da baixa concentração de pessoas na terra indígena do Trocará, outro fator que corrobora para o desaparecimento da língua é a questão da herança linguística. Um site criado pelo programa Povos Indígenas no Brasil do Instituto Socioambiental revela que a língua portuguesa tem tomado cada vez mais espaço na vida dos aldeões, ao ponto em que a maioria já a utilizaria quase que de forma exclusiva, sobretudo os mais jovens. Essa propensão à utilização do português em vez do Asuriní, somando ao fato de que os mais velhos são os que mais se comunicam na língua, pode levar o idioma a um iminente desaparecimento.

Há também uma baixa quantidade de materiais produzidos no idioma. As principais fontes incluem: livro de relatos, dicionário Asuriní-Português e artigos. Mesmo na internet há pouca informação sobre a língua. Em suma, a falta de materiais e o declínio no número de falantes é o que a torna vulnerável. Portanto, sua documentação, ou revitalização, é uma forma de dar mais visibilidade à língua.

3. REVITALIZAÇÃO

"Revitalizar é dar uma nova vida à língua" (RUBIM, 2020), como diria Altaci Rubim em sua pesquisa realizada em 2020 sobre a vitalização da língua Kokama. A autora, juntamente com outros linguistas como David Crystal, discute sobre determinadas condições que devem ser cumpridas para que o idioma seja revitalizado. Algumas dessas opções incluem: o interesse da população local para que o processo seja realizado, reaproveitar materiais existentes na língua, recursos financeiros, linguistas e outros fatores (CRYSTAL, 2005).

Especialistas como Rubim demonstram uma preocupação com a revitalização em meios sociais, com o uso recorrente do idioma e reconstruções sintáticas e até mesmo lexicais. Entretanto, o meio virtual também possibilita que vários aspectos da língua sejam documentados: não só registros escritos, mas também visuais e até mesmo sonoros.

Uma etapa que poderia ser o passo inicial para a inserção de uma língua no ciberespaço é a transposição do registro escrito para o digital, por meio de um teclado virtual. Como sabemos, cada língua possui sua própria forma de expressão escrita, por meio de letras, diacríticos e outros elementos visuais. Algumas línguas indígenas possuem um sistema gráfico distinto dos já

conhecidos atualmente, como é o caso do Cherokee, ou Tsalagi, idioma oficial dos povos originários Cherokee dos Estados Unidos, que utilizam um silabário criado especificamente para representar essa língua (CUSHMAN, 2010).

Apesar de não ser o caso do Asuriní, já que a língua também utiliza do alfabeto latino para as representações gráficas, ainda assim seria necessária uma adaptação. A utilização de diferentes teclados não se restringe à necessidade de utilizar um símbolo diferente, mas também ao conforto e à prioridade que determinadas letras têm. Tomando como exemplo duas línguas latinas, o português e o espanhol, ambas possuem os mesmos grafemas, porém estão posicionados de forma diferente. O "ñ" do espanhol pode ser replicado no teclado brasileiro, basta pressionar duas teclas: o (~) seguido da letra (n). A mesma coisa vale para o "ç" do espanhol, bastando uma sequência de teclas para ser reproduzida da mesma forma. Além disso, a disposição das letras em um teclado também está relacionada com a questão de necessidade.

O "ç" do português está presente como uma tecla única, sem a necessidade de utilizar combinações para executá-la, justamente por ser utilizada regularmente. A mesma coisa vale para o "ñ" do espanhol. Apesar de ambas possuírem a mesma escrita, ainda assim adotam teclados diferentes por uma questão de conforto.

Seguindo essa lógica, é importante salientar as características gráficas do Asuriní, para então propor um modelo condizente com a língua. Uma vogal que poderia ter uma tecla específica é o "ý" — possivelmente no lugar de alguma letra que não exista em Asuriní, como o "z" e o "x" — já que ela é muito recorrente e representa um som que não existe em português, sendo uma vogal alta e não-arredondada como o "i", mas central como o "a" (CABRAL; RODRIGUES, 2003). Porém, essa é uma decisão arbitrária, não há um consenso sobre o posicionamento ideal de cada símbolo.

A questão levantada acima é uma mera reflexão sobre as diversas possibilidades de disposição das teclas, mas ela não é o único fator que precisa ser considerado. A fim de evitar investimentos e tempo em ferramentas que são dispensáveis, como é o caso de um teclado físico, é possível utilizar do padrão brasileiro (ABNT e ABNT2) para digitar em Asuriní. Isso seria possível através de um software que permita remapear teclas, isto é, atribuir um símbolo diferente para uma determinada tecla pressionada. Há inúmeras ferramentas que podem ser utilizadas gratuitamente através de determinados sistemas operacionais. Considerando o Windows, cujo

sistema operacional é o mais amplamente utilizado pela população brasileira, é possível adquirir um software de teclado personalizável, como é o caso do Microsoft Keyboard Layout Creator, uma ferramenta gratuita disponibilizada pela própria Microsoft para Windows. Transpor o registro escrito, no entanto, não implica no ato de revitalizar a língua, apenas auxilia os usuários a trabalharem com o idioma no ciberespaço de forma mais prática.

Um meio capaz de utilizar do ciberespaço para auxiliar no processo de revitalização é o uso da internet. Ela desempenha um papel muito importante, pois com ela é possível acessar dados e compartilhar informações de qualquer parte do mundo. Essa característica, portanto, a favorece como um grande potencial para auxiliar línguas minoritárias. Além de possibilitar o acesso entre as grandes massas, também possui uma capacidade infinita de armazenagem de dados. A internet também pode ser utilizada como possibilitador para que programas e softwares possam ser baixados e executados para a tarefa de processamento de dados, já que muitos recursos estão vinculados ao serviço online.

3.1. PROCESSAMENTO DE LINGUAGEM NATURAL

Um desses recursos disponíveis para download e uso online é o Python, uma linguagem de programação capaz de executar inúmeras funções. Um artigo publicado em 2019 por Hans-Petter Halvorsen o aponta como a linguagem de programação mais utilizada, servindo para propósitos como: a criação de aplicativos, simulações, cálculos, desenvolvimento web e outros fatores (HALVORSEN, 2019).

Apesar de parecerem campos não-relacionáveis, o Python também pode ser aplicado à linguística. O nome dado ao emprego de línguas naturais — isto é, a forma de comunicação humana — no âmbito da programação, a fim de possibilitar o software a reconhecer padrões linguísticos, é Processamento de Linguagem Natural, ou PLN. Porém, o pré-processamento, podendo ser resumido como uma série de etapas prévias ao PLN, faz-se necessário para que a máquina seja capaz de compreender a língua no aspecto sintático e morfológico. Algumas dessas etapas são: sentenciação, tokenização, tokenização em subpalavras, normalização, PoS taggin e anotação de atributos morfológicos (CASELI; NUNES, 2023). Em suma, se trata de etapas fundamentais que objetivam a análise minuciosa da língua, delimitando o que são afixos, radicais,

o que é uma palavra, a recorrência com que aparecem, o que é uma frase, abreviações, dentre outros aspectos.

Ao transmitir todos esses dados para a máquina, é possível iniciar o processamento da linguagem natural. A forma com que os dados são processados se dá por meio de códigos em linguagem de programação. Tomamos como exemplo a seguinte função:

```
#Língua Asuriní do Tocantins
x = "Asuriní", “trocará”
print(x)
```

O “#” é usado para nomear coisas dentro do prompt de comando (área em que os comandos são digitados para executarem uma determinada tarefa). Ele não executa uma função, mas pode ser usado para identificar determinados segmentos. O “x” tem a função de carregar o significado do que vem após o sinal de “=”, ou seja, “x” significa ‘Asuriní’ e ‘trocará’, nesse contexto. Ele pode ser nomeado de qualquer forma, sendo “x” apenas uma exemplificação. Por fim, o exemplo também traz a função “print()”, responsável por mostrar o resultado de uma função. No caso acima, “print()” vai mostrar o resultado: “Asuriní” e “trocará”. Caso o programa seja rodado sem essa função, nada será exibido, portanto, essa função é o que dá visibilidade a determinadas partes do código.

Essas funções, além de representarem a forma como a programação é feita, também são importantes para entender o funcionamento de tradutores automáticos e outras formas de revitalização que serão trabalhadas posteriormente. Além disso, o Python pode ser útil para o desenvolvimento de revisores ortográficos, dicionários e até mesmo para a criação de inteligências artificiais.

No que se refere à aplicação do Python na criação de tradutores automáticos, está a praticidade em traduzir palavras e orações sem precisar dispensar muito tempo procurando significados em dicionários. Apesar de não ser totalmente eficaz, pois traduções automáticas são suscetíveis a erros, ainda assim pode otimizar tempo de pesquisa ao decifrar trechos de um texto ou mesmo descobrir como construir frases. Essa ferramenta, portanto, pode servir como um

método de aprendizado inicial da língua, tendo em vista a ausência de materiais didáticos em Asuriní do Trocará.

Justamente por ser uma tradução não revisada, é importante haver um dicionário digital, pois é nele onde será possível encontrar os verbetes com todas as acepções possíveis, isto é, os significados que uma determinada palavra pode ter. Apesar de já existir um dicionário em Asuriní-Português (2003) — como o que está sendo utilizado como material bibliográfico para esta pesquisa — a transposição disso para a linguagem de programação poderia facilitar em alguns aspectos: a versatilidade e a praticidade. Um dicionário publicado, como aquele utilizado como material bibliográfico para esta pesquisa, não pode ser editado, diferentemente de um que for feito em Python. Qualquer usuário que disponha do código consegue alimentar o programa e complementar o dicionário com novas palavras ou acepções. Além disso, sua praticidade se deve à estrutura de como ele é trabalhado dentro da programação. Em suma, o usuário poderia digitar a palavra desejada e o programa exibiria o verbete que contém aquela palavra, juntamente com as acepções.

Tendo em vista que para o funcionamento da tradução automática é necessário alimentar a máquina com o máximo de informações possíveis sobre o idioma, o mesmo ato também corrobora para que a revisão ortográfica e a inteligência artificial sejam possíveis. Uma vez que o programa recebe informações o suficiente sobre as estruturas gramaticais da língua, ele consegue, por meio de um input que permita o usuário digitar as palavras ou orações desejáveis, verificar se o que foi escrito está de acordo com as informações que lhe foram atribuídas.

A inteligência artificial (IA) é a área da ciência da computação que estuda como as máquinas conseguem simular a inteligência humana (ARTASANCHEZ; JOSHI, 2020). Apesar de ser possível reproduzi-la em Python, apresenta um grau de complexidade muito elevado. É importante salientar que há diferentes ramos dentro da IA, cada uma responsável por executar alguma tarefa. No caso da revitalização linguística, os chatbots — inteligência artificial voltada para a conversação — podem ser úteis para esclarecer dúvidas sobre idiomas. Apesar de já existirem chatbots com grande conhecimento em linguagens, como o ChatGPT, estes recursos ainda assim foram alimentados com poucas informações acerca de línguas minoritárias, portanto, são incapazes de responder perguntas relacionados ao assunto.

3.2. PRESENÇA NO CIBERESPAÇO

Revitalizar não se limita ao uso de programação, todavia. A criação de sites é um exemplo de recurso capaz de auxiliar na documentação de línguas, isso porque se trata de um ambiente online acessível por meio da internet e com capacidade de manipulação das mais variadas formas. É possível utilizá-lo como um espaço para execução de softwares, como o prompt online do Python (existem sites online que possibilitam o usuário acessá-lo sem a necessidade de instalação) e até mesmo como repositório de arquivos. Essa função de armazenagem é útil não só para disponibilizar materiais de forma acessível, mas também para ser utilizada como corpus para processamento de texto.

Corpus, no plural corpora, pode ser definido como uma coletânea de textos naturais, escolhidos para caracterizar um estado ou variedade de linguagem (SARDINHA, 2004). Esses textos utilizam de fontes autênticas, excluindo geradores de texto e métodos relacionados, para analisar características como: vocabulário, sintaxe, frequência ou até mesmo concordância. Podem ser encontrados de forma online ou criados por meio de algum programa. A utilidade de tal meio se deve à versatilidade em identificar estruturas gramaticais de forma mais precisa, facilitando no estudo do idioma e até mesmo na tradução. Para demonstrar a utilidade de um corpus como ferramenta passível de revitalização, será apresentado um exemplo, proveniente do software AntConc (disponível de forma gratuita na internet e sem a necessidade de instalação).

Figura 1 – Análise de corpus no AntConc

fillings in igneous	rocks.	Uses: Limestone
found in igneous	rocks	such as
Occurs in igneous	rocks	like granites,
in felsic igneous	rocks	such as
other felsic igneous	rocks	and in
commonly in igneous	rocks	like granites,
and intermediate igneous	rocks	and forms
rich volcanic igneous	rocks	like rhyolite
sedimentary and metamorphic	rocks.	It is
and gneiss metamorphic	rocks.	It is
occurs in metamorphic	rocks	and sediments.
of many metamorphic	rocks	and immature
other related metamorphic	rocks.	Uses: Quartz
and clastic sedimentary	rocks.	Uses: Commercially,
some immature sedimentary	rocks.	Uses: Mica
present in sedimentary	rocks	like sandstone
in other sedimentary	rocks.	Calcite is

Fonte: própria autora

Os excertos acima foram retirados de um artigo sobre minerais, posteriormente anexado ao programa. Na interface do AntConc, foi digitado “rocks”, que significa “rochas/pedras”, como a palavra de referência a ser escaneada pelo documento. É possível verificar que “rocks” sempre é acompanhada por um adjetivo, variando entre: ígneo, metamórfico e sedimentar, precedendo a palavra principal. A utilidade disso é verificar palavras que normalmente estão associadas à outras, além da posição em que aparecem. Este exemplo, no entanto, foi retirado apenas de um texto, o que significa que os resultados acima obtidos possam variar com aglomerados de textos, ou seja, os corpora.

Em um primeiro momento, é possível inferir que os corpora não são necessários, tendo em vista que existem tradutores e até mesmo motores de busca para isso (como o Google). É possível digitar “rocks” e verificar quais palavras antecedem e precedem o termo. Entretanto, há casos em que tanto os tradutores como os motores de busca não irão fornecer os resultados

esperados, com múltiplas opções de tradução e resultados disponíveis. A função “collocates” dos corpora, por exemplo, permite com que o usuário identifique palavras de uma classe gramatical específica em qualquer posição, com base na palavra digitada (permitindo filtrar os resultados apenas para adjetivo, por exemplo). Essa não é, entretanto, a única vantagem. Também é possível utilizar os corpora para descobrir palavras. Tomamos como exemplo a seguinte situação: sabe-se que existem inúmeros pratos culinários, cada um com um nome e características distintas. Caso um usuário esteja à procura de uma receita culinária específica em inglês, porém sem lembrar o nome, apenas com a informação de que o prato leva a palavra “peixe” no nome, terá que utilizar motores de busca com perguntas semelhantes a “pratos com peixe” ou até mesmo “pratos que levam peixe no nome” para encontrá-lo. Os resultados, no entanto, podem variar drasticamente (tendo em vista a infinidade de pratos com essa característica), muitos até sem obedecer ao comando que lhe fora imposto. Caso o prato selecionado seja o Gelfite Fish (prato judaico com peixe cozido), é possível que o motor de busca sequer o mencione nas principais buscas e sites, mas sim receitas que levam peixe em sua composição. O uso de corpora, nesse caso, como o COCA do inglês, permitiria localizar esse prato com facilidade, bastando digitar a palavra “fish” e buscar por um collocate específico que acompanha essa palavra, no caso, um substantivo à esquerda de “fish”.

Em suma, o uso de corpora permite uma análise técnica sobre a morfossintaxe de uma língua, permitindo uma maior precisão na colocação das palavras. Esse benefício, porém, não é acessível a todos os idiomas, e é por esse motivo que a criação de aplicativos como AntConc são fundamentais, para que sejam alimentados a trabalhar com línguas indígenas como o Asuriní. Dada a explicação do porquê os corpora são relevantes, é importante frisar que a eficácia dos mesmos está diretamente relacionada com a quantidade de documentos que são utilizados como base, e é por esse motivo que disponibilizar documentos dos mais variados assuntos de forma centralizada poderia facilitar no processo de formação de um corpus do Asuriní mais completo. O site, portanto, poderia servir como um meio centralizador de documentos para esse fim.

Para compreender o funcionamento de um site, primeiramente, é necessário que haja um Sistema de Gerenciamento de Conteúdo (CMS), cuja função é fazer as edições e personalizações (pode ser resumido como a parte criativa). Em seguida, é preciso que ele seja disponibilizado na web, e é por meio do serviço de hospedagem que ele é feito. O site também precisa de um

domínio, ou seja, um endereço que permita localizá-lo (ex: www.google.com). Todos esses procedimentos podem ser feitos de forma gratuita, apesar de serem restritos em certas funcionalidades e possuírem desvantagens, tais como: anúncios, baixa segurança, pouco espaço de armazenamento, dentre outros.

Ainda que seja feito na versão gratuita, a existência de um espaço online dedicado ao Asuriní corrobora para a visibilidade da língua e serve como um meio capaz de centralizar as propostas de revitalização previamente mencionadas. Tendo explicado todos os pontos, é importante, agora, entender a metodologia desta pesquisa.

4. METODOLOGIA

A função das ferramentas de auxílio à revitalização discutidas nas seções anteriores — isto é, a criação de um site para o Asuriní, um teclado adaptado para a língua, a existência de um dicionário digital, um tradutor automático, corretores ortográficos, corpora e o uso de chatbots — é de democratizar o conhecimento acerca da língua indígena dos Asurinís do Tocantins, permitindo com que qualquer pessoa possa acessar os materiais e informações acerca do idioma por meio da internet. Esses recursos, portanto, podem servir como potenciais ferramentas de estudo para profissionais que estejam a par da iniciativa de revitalização, podendo ser utilizados para o aprendizado da língua ou mesmo como aliados na tarefa de decifração de novos termos e significados.

A pesquisa se propõe em demonstrar de forma prática como a língua pode ser trabalhada no ciberespaço, dando ênfase na aplicação de ferramentas que serão mais relevantes para a tarefa de revitalização, como o tradutor automático, a criação de um modelo de site específico do Asuriní do Tocantins e também de um dicionário eletrônico, por meio da utilização do Python e CMS. O motivo para essa delimitação pode ser justificado pelo próprio objetivo da pesquisa, cujo foco não é necessariamente de apresentar todas as ferramentas já desenvolvidas, um site ativo ou qualquer outro recurso já finalizado, mas sim de poder mostrar que essas propostas são possíveis e de quais formas elas podem ser elaboradas. Portanto, conforme as ferramentas forem apresentadas, serão dadas explicações sobre a estrutura aplicada, a fim de facilitar para que outras pessoas possam replicar ou se basear nos dados disponibilizados.

Tendo em vista que determinados recursos são dependentes de outros, além do grau de complexidade de cada ferramenta, será utilizado um critério lógico para o ordenamento dos resultados, a fim de facilitar no entendimento. Nesse caso, devido à importância do site como principal ferramenta que reunirá as demais propostas, será o primeiro a ser desenvolvido.

5. WORDPRESS E O SITE DO ASURINÍ DO TROCARÁ

A ferramenta utilizada como referência para a criação do site, nesta pesquisa, é o WordPress, tanto como CMS quanto para a hospedagem. A não obrigatoriedade de possuir conhecimento em HTML e CSS para a personalização e geração do conteúdo, ainda que possível por meio destes, o torna uma plataforma acessível a todos, e é por isso que ela foi utilizada como referência, pois o intuito é que qualquer pessoa possa utilizar dos conhecimentos aqui obtidos para criar suas próprias versões. O domínio não pode ser alterado na versão gratuita, mas serve como protótipo para demonstrar na prática o ordenamento de cada funcionalidade. A imagem abaixo é uma captura de tela da interface do WordPress, onde é possível localizar as ferramentas de criação e o conteúdo a ser exibido (ao meio).

Figura 2 – Site do Asuriní do Trocará

Fonte: própria autora

Considerando que a revitalização do idioma é a prioridade, esse *layout* representaria apenas uma página específica dentro do site, voltada para o idioma. O botão "Voltar à página principal – Os Asurinís", no canto superior esquerdo, foi pensado para retornar à página que apresentaria sobre a comunidade indígena como um todo. Tendo em vista que os detalhes acerca do povo Asuriní não são o foco da pesquisa, mas sim o idioma, a página inicial não foi desenvolvida, servindo apenas como referencial para entender a forma de acesso do exemplo acima, que, no caso, seria por meio de um hiperlink.

A interface principal possui um título e um espaço dedicado à apresentação do idioma, juntamente com um carrossel de imagens (pensadas para mostrar a grafia do idioma e derivados) e também botões, cuja função é redirecionar o usuário às demais páginas que envolvem a utilização do Python. A exceção, no entanto, seria o último hiperlink: "arquivos de áudio", pensado como um espaço para armazenar gravações de voz, vídeos e quaisquer elementos que estejam relacionados à fala. Ainda que a pesquisa não abarque ferramentas de revitalização voltadas a esse meio, é importante salientar a importância do registro fonético, para que a comunicação não seja restrita apenas ao meio escrito.

O footer, ou rodapé, não será apresentado nesta primeira versão, por ser uma ferramenta pouco útil para as propostas de revitalização, já que, normalmente, tem a função de retomar links de outras páginas, apresentar telefone/e-mail de contato, dentre outras opções. O que pode ser adicionado na página sobre o idioma é o teclado já personalizado pelo Microsoft Keyboard Layout Creator. Entretanto, por se tratar de uma ferramenta que pode ser facilmente acessada e personalizada de outras formas, foi desconsiderada para o corpo do site. Uma vez que a interface esteja pronta, faz-se necessário explicar o funcionamento do Pyhton para o desenvolvimento das páginas: "tradutor automático", "dicionário digital", "revisor ortográfico" e "chatbot". Sobre a página "corpora", ela pode tanto conter uma seção voltada para armazenar arquivos de textos diversos, como também uma parte dedicada à elaboração de um aplicativo de processamento de corpus, como o AntConc, por exemplo.

6. CRIAÇÃO DO DICIONÁRIO DIGITAL COM PYTHON

Para que o Python consiga executar uma tarefa, é necessário que sejam utilizados vários elementos, tais como: funções, variáveis, operadores, indentações e outros. Eles podem ser

pensados como comandos que possibilitam a máquina a compreender e executar o que está sendo solicitado pelo usuário. Para exemplificar, será utilizado um código que possibilita a criação de um dicionário digital do Asuriní.

Figura 3 – Dicionário digital do Asuriní

Fonte: própria autora

No código acima, é possível verificar a utilização de aspas, cerquilha, parênteses e outros elementos. Tendo em vista o grau de complexidade em explicar detalhadamente sobre cada um deles, além de demandar espaço para a explicação dos mesmos, será abordado apenas o essencial, o suficiente para que seja possível compreender como o código geraria um dicionário.

A primeira parte é entender o uso da cerquilha (#), que pode ser resumida como um nomeador de elementos, porém sem fazer parte da execução do código. No caso do exemplo acima, foi usada para separar as categorias de palavras do dicionário — palavras com "A" e "M" — e também para definir que aquele código se refere ao dicionário da língua, visível logo na primeira linha. Os elementos em verde, acompanhados por aspas simples ou duplas, representam todo o texto que será exibido no resultado final (ou que pode ser exibido, a depender das condições), assim que o botão "*Run*" for pressionado. Em amarelo, é possível localizar, primeiramente, a função "input()". Ela é responsável por adicionar uma linha digitável que, no caso do dicionário, sucede o texto "Digite uma palavra em Asuriní:" (linha 2). Ainda na mesma

linha, ao final, há o método de string “.lower()”, cuja finalidade é de converter tudo o que for digitado para letras minúsculas (intuito de padronizar os resultados). Por fim, a utilização do ".join()” (linha 13), serve como uma concatenadora de textos, responsável por exibir todas as acepções da palavra digitada.

A combinação de "if”, “in” e “else”, em laranja, diz respeito a uma condição que, caso seja cumprida, algo aconteceria, do contrário, outro resultado seria exibido (if, in e else, respectivamente). No caso do dicionário, se a palavra digitada pelo usuário (if) estiver no dicionário (in), será exibido (print) a mensagem “A palavra ‘palavra’ pode significar: ‘acepções’, caso contrário (else), será exibido (print): “Palavra não encontrada”. Logo, é possível perceber que a função “print()” tem como objetivo mostrar textos que estão inclusos no código, a depender da circunstância.

Mesmo sem explicar a utilização de colchetes, chaves e afins, é possível entender, com a explicação acima, como o resultado desse código resulta em um dicionário, ainda que simplificado e específico do Asuriní para o português. Assim, o intuito do código é mostrar o funcionamento essencial de um dicionário, permitindo que o usuário alimente o código com mais palavras e significados, aprimorando a ferramenta para que possa ser útil a todos.

7. TRADUTOR AUTOMÁTICO

Algumas partes do código acima podem ser reutilizadas para a criação de um tradutor automático, como é possível verificar nas imagens abaixo.

Figura 4 – Primeira parte do código do tradutor automático

main.py

```
texto = input("Tradutor automático do asuriní: ").lower()
dicionário = {
    "a'é": [("esse", "d"), ("essa", "d"), ("isso", "d")],
    "oré": [("nós", "p")],
    "re'ýsa": [("parente", "s")],
    "tým": [("enterrar", "v", "enterr")],
    "e'ýj": [("coçar", "v", "coç")],
    "a'ýj": [("semente", "s"), ("azedo", "a")],
    "akwáp": [("índio brabo", "s"), ("pêlos pubianos", "s")],
    "mirikatáp": [("namoro", "s")]
}
afixos = {
    "o": [("ele", "prefixo")],
    "a": [("eu", "prefixo")]
}
def traduzir_verbo_com_prefixo(palavra, dicionário):
    if len(palavra) > 1 and palavra[0] in ["a", "o"]:
        verbo = palavra[1:]
        for chave, valores in dicionário.items():
            if chave == verbo and valores[0][1] == "v":
                radical = valores[0][2]
                if palavra[0] == "a":
                    return radical + "o"
```

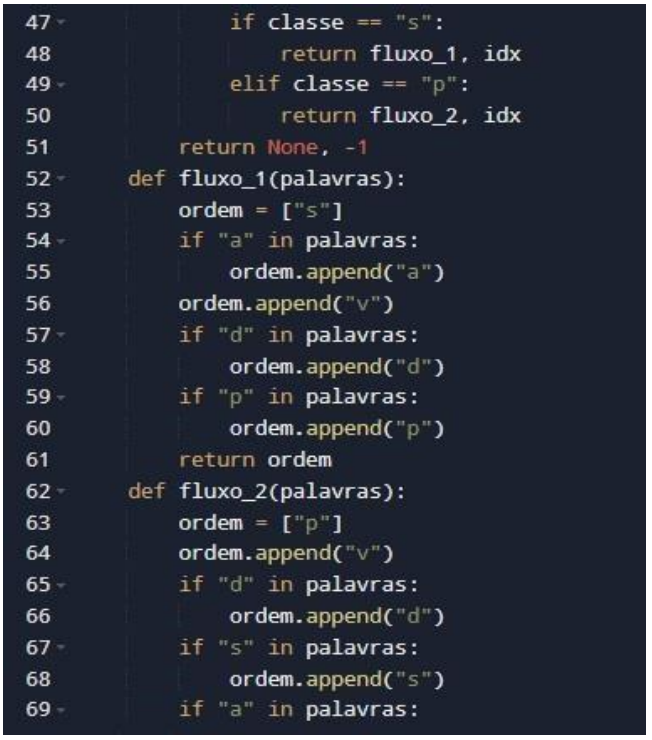
Fonte: própria autora

Figura 5 – Segunda parte do código do tradutor automático

```
                elif palavra[0] == "o":
                    return radical + "a"
    return None
def encontrar_acepção_alternativa(palavra, classe_esperada):
    if palavra in dicionário:
        for tradução, classe in dicionário[palavra]:
            if classe == classe_esperada:
                return tradução, classe
    return None, None
def traduzir_com_prefixos(texto, dicionário, afixos):
    palavras = texto.split()
    resultado = []
    acepções_palavras = []
    def encontrar_acepção(palavra):
        tradução_verbo = traduzir_verbo_com_prefixo(palavra, dicionário)
        if tradução_verbo:
            return tradução_verbo, "v"
        if palavra in dicionário:
            return dicionário[palavra][0][0], dicionário[palavra][0][1]
        return palavra, None
    def determinar_fluxo(palavras):
        for idx, palavra in enumerate(palavras):
            tradução, classe = encontrar_acepção(palavra)
```

Fonte: própria autora

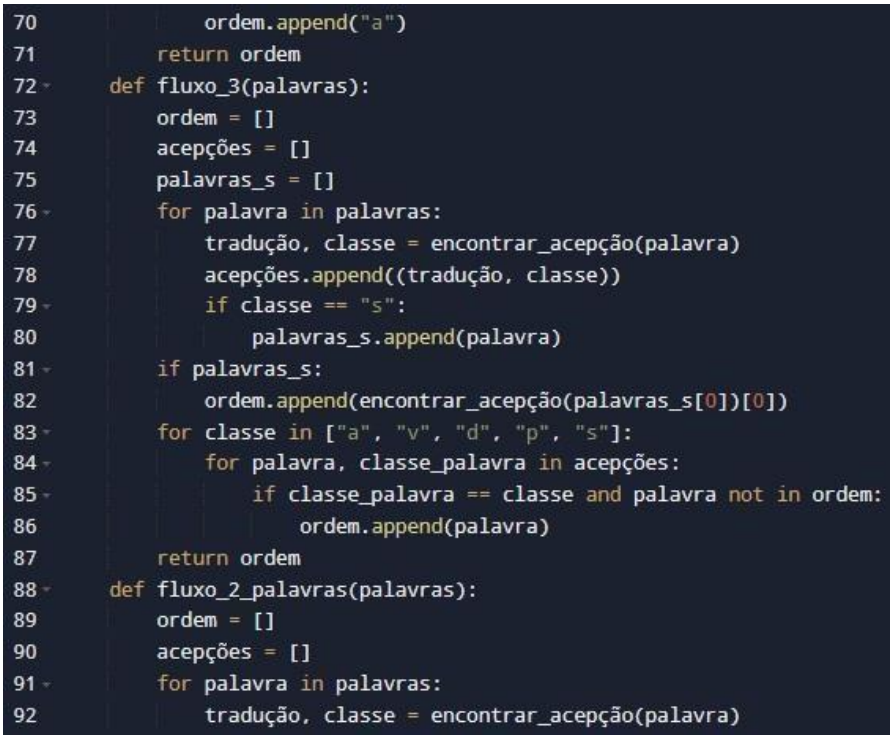
Figura 6 – Terceira parte do código do tradutor automático



```
            if classe == "s":
                return fluxo_1, idx
            elif classe == "p":
                return fluxo_2, idx
        return None, -1
    def fluxo_1(palavras):
        ordem = ["s"]
        if "a" in palavras:
            ordem.append("a")
        ordem.append("v")
        if "d" in palavras:
            ordem.append("d")
        if "p" in palavras:
            ordem.append("p")
        return ordem
    def fluxo_2(palavras):
        ordem = ["p"]
        ordem.append("v")
        if "d" in palavras:
            ordem.append("d")
        if "s" in palavras:
            ordem.append("s")
        if "a" in palavras:
```

Fonte: própria autora

Figura 7 – Quarta parte do código do tradutor automático



```
            ordem.append("a")
        return ordem
    def fluxo_3(palavras):
        ordem = []
        acepções = []
        palavras_s = []
        for palavra in palavras:
            tradução, classe = encontrar_acepção(palavra)
            acepções.append((tradução, classe))
            if classe == "s":
                palavras_s.append(palavra)
        if palavras_s:
            ordem.append(encontrar_acepção(palavras_s[0])[0])
        for classe in ["a", "v", "d", "p", "s"]:
            for palavra, classe_palavra in acepções:
                if classe_palavra == classe and palavra not in ordem:
                    ordem.append(palavra)
        return ordem
    def fluxo_2_palavras(palavras):
        ordem = []
        acepções = []
        for palavra in palavras:
            tradução, classe = encontrar_acepção(palavra)
```

Fonte: própria autora

Figura 8 – Sexta parte do código do tradutor automático

```
            acepções.append((tradução, classe))
            if classe == "s" or classe == "p":
                ordem.append(classe)
        if acepções[0][1] == acepções[1][1]:
            palavra_1 = palavras[0]
            palavra_2 = palavras[1]
            tradução_1, classe_1 = encontrar_acepção(palavra_1)
            tradução_2, classe_2 = encontrar_acepção(palavra_2)
            if classe_1 == classe_2:
                tradução_alternativa_2, _ = encontrar_acepção_alternativa(palavra_2, "a")
                if tradução_alternativa_2:
                    return [(tradução_1, classe_1), (tradução_alternativa_2, "a")]
                else:
                    tradução_alternativa_1, _ = encontrar_acepção_alternativa(palavra_1, "a")
                    return [(tradução_alternativa_1, "a"), (tradução_2, classe_2)]
        else:
            return acepções
    if len(palavras) == 1:
        tradução, _ = encontrar_acepção(palavras[0])
        return tradução
    fluxo, index = determinar_fluxo(palavras)
    if fluxo:
        palavras_reordenadas = palavras[index:] + palavras[:index]
```

Fonte: própria autora

Figura 9 – Sexta parte do código do tradutor automático

```
        ordem_fluxo = fluxo([encontrar_acepção(p)[1] for p in palavras_reordenadas])
        if len(palavras) == 2:
            palavras_traduzidas = fluxo_2_palavras(palavras)
            ordem_prioridade = {"s": 1, "p": 1, "a": 2, "v": 3}
            palavras_traduzidas.sort(key=lambda x: ordem_prioridade.get(x[1], 4))
            resultado = [tradução for tradução, _ in palavras_traduzidas]
        else:
            if len(set([encontrar_acepção(p)[1] for p in palavras])) < len(palavras):
                palavras_traduzidas = fluxo_3(palavras)
                resultado = palavras_traduzidas
            else:
                for classe in ordem_fluxo:
                    for palavra in palavras_reordenadas:
                        tradução, palavra_classe = encontrar_acepção(palavra)
                        if palavra_classe == classe and tradução not in resultado:
                            resultado.append(tradução)
                            break
    return " ".join(resultado)
try:
    traduzido = traduzir_com_prefixos(texto, dicionário, afixos)
    print(traduzido)
except Exception as e:
    print(f"Erro: {e}")
```

Fonte: própria autora

Apesar de ser uma versão simplificada e ainda incompatível com a gramática do Asuriní em determinados aspectos, o exemplo acima mostra como o código pode ser estruturado. Primeiramente, é importante atribuir às palavras do dicionário uma classe gramatical, pois esse critério facilita no ordenamento do resultado. Entretanto, um mesmo verbete pode ter múltiplas acepções com classes gramaticais distintas. Isso pode ser observado por meio da palavra “a’ýj”, cujas definições variam entre substantivo, adjetivo e verbo (classe gramatical não utilizada no dicionário ilustrativo). Para prevenir equívocos, faz-se necessário estabelecer condições para que o programa saiba o que fazer em cada situação.

Sabemos que na língua portuguesa é comum a ordem sintática S.V.C (sujeito, verbo e complemento), ou seja, a estrutura das orações segue um formado esperado. Para que o posicionamento das palavras do input siga uma ordem semelhante a essa, foi estabelecida uma série de condições para cada classe gramatical encontrada no input, sendo observada por meio dos fluxos (imagem 3 e 4). Nessas linhas de código, é possível verificar que cada fluxo é determinado a partir da primeira palavra do input, variando entre substantivo “s” e pronome “p”. A partir dessas classes gramaticais, as outras serão posicionadas, por ordem de prioridade, ignorando aquilo que não for aplicável. Essa condição é importante porque não há ordem fixa das palavras em Asuriní, então posicioná-las de forma mais próxima à estrutura do português facilita no entendimento da oração, evitando estruturas que não são usuais ou até inexistentes na língua portuguesa, como: madura caiu maçã (A.V.S).

A fim de evitar falhas na tradução por conta de estruturas que não estejam de acordo a lógica dos fluxos, três linhas de código foram adicionadas (observável na imagem 5). Essa condição específica permite com que a ocorrência de uma só palavra no input seja possível, traduzindo-a com base na primeira acepção do dicionário. Há outras formas de complementar isso, adicionando novas linhas de código para mostrar todas as acepções de uma palavra, como já visto anteriormente no dicionário digital, por meio da seguinte linha de código: print(f" {', '.join(acepções)}").

Caso o input comece por uma palavra que não seja um substantivo ou um pronome, o programa irá verificar a palavra seguinte até encontrar a primeira ocorrência dessas duas classes, posicionando-as, então, no início. Após esse processo, ele seguirá a ordem dos fluxos 1 e 2, a depender de qual tenha sido a primeira palavra a ser identificada.

Existem frases em português que não começam com pronome ou substantivo, porém, para abordar cada exceção, seria necessário um tradutor muito mais robusto para lidar com todas as particularidades da língua. Dito isso, foi desconsiderado a passividade das orações e outros elementos com ordens variáveis.

Outro elemento importante pode ser identificado nas duas primeiras imagens, onde o código aborda um método de conjugação de verbos. De forma resumida, ele faz uma checagem inicial das palavras do input e, caso comece com as vogais “a” e “o” — cuja finalidade é determinar o sujeito, sendo “a” para a primeira pessoa do singular e “o” para a terceira pessoa do singular” — o código analisará o restante da palavra, excluindo esse prefixo. Caso seja compatível com um verbo presente no dicionário, essa palavra irá selecionar o radical (também presente no dicionário ilustrado) e acoplar o prefixo correspondente à pessoa a quem se refere aquela ação. Para facilitar no entendimento, tomamos como exemplo a palavra “tým”, que significa “enterrar”. Caso essa palavra apareça no input como “otým”, o tradutor irá identificar que essa palavra é um verbo e irá traduzi-la não como “enterrar”, mas sim como “enterra”, já que essa vogal marca a terceira pessoa (ele enterra).

A explicação acima sobre o funcionamento do conjugador é essencial para entender como a máquina é capaz de analisar minuciosamente uma palavra. Com base nas informações linguísticas do Asuriní trabalhadas ao longo do texto, é possível observar que uma só palavra pode abarcar diversos significados e classes gramaticais, podendo até mesmo sintetizar um enunciado completo — a depender dos afixos presentes — como pode ser observado no seguinte exemplo: “okájihi”. Podemos fragmentar essa estrutura em algumas partes: “o”, “káj” e “ihi”. A lógica do prefixo “a” já foi previamente adaptada no conjugador, marcando a presença do sujeito. “Káj” é um verbo e significa ‘queimar’. Por fim, o sufixo “ihi”, que é a negação. Em suma, uma possível tradução para “okájihi” seria: “ele não queima”. Portanto, elaborar um tradutor complexo que consiga analisar cada elemento de uma palavra é essencial, pois a língua é repleta de aglutinações.

Voltando para a explicação do tradutor automático, também é importante salientar que determinadas condições foram estabelecidas para lidar com a ocorrência de classes gramaticais repetidas. Caso dois substantivos apareçam juntos, em um input com dois elementos, o tradutor irá verificar se existe alguma acepção diferente para alguma dessas palavras. Se for aplicável, ele

fará a substituição da primeira acepção do dicionário por uma que seja diferente em termos morfológicos. Vamos supor que uma frase hipotética como “a’ýj a’ýj” apareça no input. Em tese, deveria ser traduzida como “semente semente”, o que não faz muito sentido. Porém, com a condição estabelecida no código acima, o resultado será “semente azedo”, melhorando no entendimento do enunciado. Ainda assim o código é imperfeito nesse sentido, pois não há concordância de gênero, o que significa que também precisará de ajustes para que a tradução seja feita de maneira mais precisa.

Essa noção de repetição de classes gramaticais, talvez, tenha sido o maior problema em toda a elaboração do código. Explicar para a máquina que a coexistência de duas classes é possível e que precisam estar posicionadas de forma lógica em português foi, provavelmente, o maior desafio. O fluxo 3, observável na imagem 4, é uma função que permite com que o programa identifique a presença de mais de uma classe gramatical e, caso se aplique, ele irá checar a primeira ocorrência do substantivo “s”, para então posicioná-lo no início do resultado. Uma vez feito isso, as demais palavras seguirão uma ordem pré-estabelecida, ignorando aquilo que não for aplicável. Esse funcionamento permite com que um dos sujeitos seja tratado como sujeito da oração, enquanto o outro ocuparia a função sintática de objeto.

Com base em tudo o que foi apresentado, é possível verificar uma série de inconsistências e limitações no tradutor automático. Fazer com que a máquina seja capaz de trabalhar com estruturas mais amplas do que a de uma oração, a concordância de gênero e número, além de outras características, é indispensável para que possa ser considerado um verdadeiro tradutor automático do Asuriní. Entretanto, esse protótipo pode servir de base para a criação de outros tradutores, ou até mesmo como modelo inicial para que aperfeiçoamentos sejam feitos no próprio código disponibilizado. Com a proposta dessa pesquisa, os dados em Python estariam disponíveis no site do Asuriní na aba “Tradutor Automático”, para que possam ser baixados e utilizados pelos usuários.

7. CONSIDERAÇÕES FINAIS

Ainda que não tenha sido elaborado um chatbot do Asuriní, é possível entender como seria possível fazer isso utilizando o Python, já que o tradutor automático mostra o quão precisa a máquina consegue ser com a quantidade de informações agregadas. A razão para esse recurso não

ter sido desenvolvido se deve ao alto nível de complexidade de um chatbot, cuja estrutura, ainda que de forma simplificada, é detalhada o suficiente a ponto de estender o trabalho de conclusão de curso em algumas páginas, sem contar com o tempo necessário para desenvolver um código funcional, mesmo que de forma simplificada.

Esse, no entanto, não foi o único elemento a ser desconsiderado na parte prática da pesquisa. A mesma coisa ocorre com o revisor ortográfico da língua que, assim como o chatbot, também utilizaria o Python. Por ser algo complementar, foi deixado apenas como uma sugestão de ferramenta que também pode ser criada utilizando o mesmo recurso do tradutor automático e do dicionário eletrônico, ademais do chatbot.

Como a ideia da pesquisa era apresentar um leque de possibilidades utilizando a diversidade dos computadores, internet e quaisquer recursos digitais para auxiliar na revitalização do idioma, e depois provar de forma objetiva como seria a execução disso, algumas dessas ferramentas foram apenas citadas e debatidas como potenciais elementos de documentação e auxílio no trabalho de linguistas, professores e quaisquer profissionais que possam estar relacionados à causa.

A razão para ter sido priorizada a criação do site, do tradutor automático e do dicionário digital se deve pela relevância que tais fatores têm para o estudo do idioma, mesmo para aqueles que não possuem conhecimento prévio do Asuriní. Primeiramente, caso alguém queira acessar materiais, fazer pesquisas sobre a língua ou qualquer outro motivo, é necessário que haja um local para acessar tais informações, e é por isso que o site foi o primeiro a ser desenvolvido, por ser aquele que tem a função de centralizar todas as informações relevantes e materiais referentes ao Asuriní. Portanto, disponibilizar o tradutor automático no site é uma forma de democratizar o conhecimento da língua, pois mesmo alguém que não tenha conhecimento prévio do idioma, o mesmo consegue entender os textos originais e aprender novas palavras e frases com auxílio dessa ferramenta. A mesma coisa vale para o dicionário, por se tratar de uma ferramenta muito útil de aprendizado. Permitir com que esses materiais sejam disponibilizados na web e com a possibilidade de serem aperfeiçoados por outras pessoas é o que o torna tão importante.

Por fim, é importante frisar que mais recursos também poderiam ter sido englobados nessa pesquisa, além de refinamentos às técnicas aqui apresentadas. Tal trabalho pode ser utilizado como base para que estudos mais aprofundados sobre o tema possam ser desenvolvidas

futuramente. Esta é, portanto, uma pesquisa que visa servir de base para que outros trabalhos possam dar continuidade ao tema. Além disso, tudo o que foi visto aqui pode ser aproveitado para auxiliar na tarefa de revitalização de quaisquer outras línguas. A utilização do Asuriní do Tocantins foi um modelo para a aplicação de tais meios, por ter sido uma língua indígena de meu conhecimento.

REFERÊNCIAS BIBLIOGRÁFICAS

CABRAL, Ana Suelly Arruda Câmara; RODRIGUES, Aryon Dall'igna. **Dicionário Asuriní do Tocantins-Português**. Belém: UFPA/IFNOPAP; Brasília: UnB/IL/LALI, 2003.

CABRAL, Ana; SILVA, Ariel; MARTINS, Daniella; LOPES, Jorge; CARVALHO, José; SOUSA, Suseile. **Esboço Gramatical do Asuriní do Tocantins**. Brasília: UnB/IL/LALI, 2011.

Asurini do Tocantins | Povos Indígenas no Brasil. Disponível em: https://pib.socioambiental.org/pt/Povo:Asurini_do_Tocantins#L.C3.ADngua. Acesso em 30 out. 2024.

CRYSTAL, David. **La Revolución del Lenguaje**. Madrid: Alianza Editorial, 2005.

SARDINHA, Tony Berber. **Lingüística de Corpus**. São Paulo: Manole, 2004.

LIMBERGER, Bernardo; KÜRSCHNER, Sebastian; ALTENHOFEN, Cléo; MOZZILLO, Isabella. **Línguas Minoritárias**. Rio Grande do Sul: Revista Linguagem & Ensino, 2020.

Os indígenas no Censo 2022. Disponível em: https://educa.ibge.gov.br/criancas/brasil/nosso povo/22324-os-indigenas-no-censo-2022.html. Acesso em 30 out. 2024.

ARTASANCHEZ, Alberto; JOSHI, Prateek. **Artificial Intelligence with Python**. Birmingham: Packt Publishing, 2020

CASELI, Helena de Medeiros; NUNES, Maria das Graças Volpe. **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**. Califórnia: Creative Commons, 2023.

Línguas Indígenas dos Países Ibero-Americanos | Dados e Gráficos. Disponível em: https://oei.int/pt/. Acesso em 16 dez. 2024.

WARSCHAUER, Mark. **Technology and Indigenous Language Revitalization: Analyzing the Experience of Hawai'i**. Canadian Modern Language Review, 1998.

Python Programming. Disponível em: https://www.halvorsen.blog/documents/programming/python/python.php. Acesso em 16 dez. 2024.

CUSHMAN, Ellen. **The Cherokee Syllabary from Script to Print**. Michigan State University, 2010.

RUBIM, Altaci Corrêa. **A Vitalização da Língua Kokama: Além das Fronteiras Entre o Brasil e Peru**. Brasília: LIP/IL/UnB, 2020.