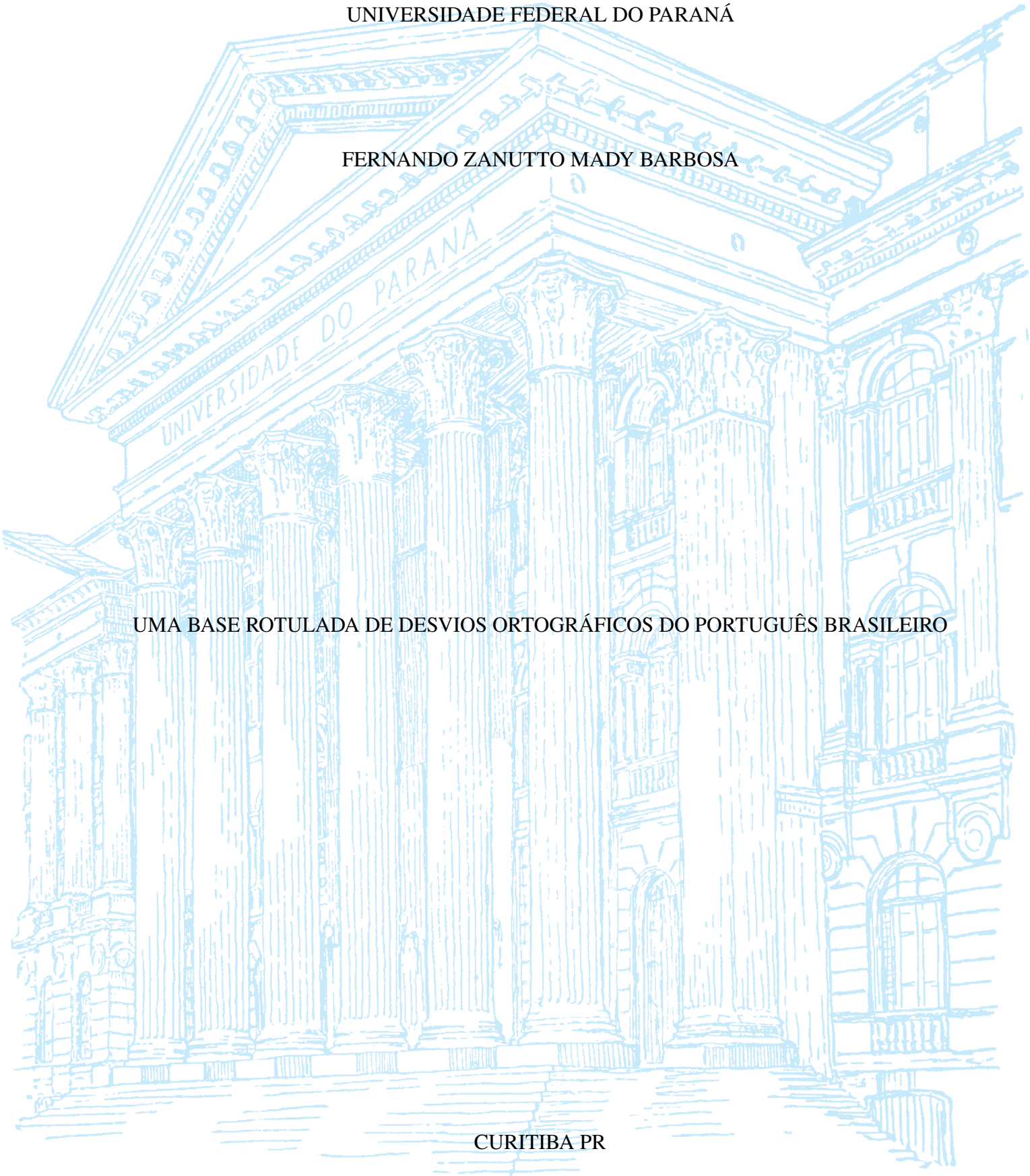
UNIVERSIDADE FEDERAL DO PARANÁ

FERNANDO ZANUTTO MADY BARBOSA



UMA BASE ROTULADA DE DESVIOS ORTOGRÁFICOS DO PORTUGUÊS BRASILEIRO

CURITIBA PR

2023

FERNANDO ZANUTTO MADY BARBOSA

UMA BASE ROTULADA DE DESVIOS ORTOGRÁFICOS DO PORTUGUÊS BRASILEIRO

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Fabiano Silva.

Coorientador: Adelaide H.P. Silva.

CURITIBA PR

2023

A minha mãe, pelo intenso apoio durante minha jornada acadêmica.

AGRADECIMENTOS

A Deus, por me capacitar de saúde física e mental para ultrapassar os desafios encontrados ao longo do curso.

Aos meus orientadores, Fabiano Silva e Adelaide Silva, pelo suporte neste estudo e pela paciência em lidar com a minha excessiva preocupação.

A minha mãe, Marcia Zanutto, por sempre me impulsionar a continuar perseguindo meus objetivos e ser uma ouvinte paciente das minhas lamentações.

A minha companheira, Maria Clara de Campos Salik, por seu otimismo e confiança na minha capacidade. Além da ótima formação, minha graduação me possibilitou conhecer essa pessoa especial que agradeço a Deus por ter ao meu lado.

Por fim, aos muitos que me incentivaram a finalizar essa jornada que seria impossível sozinho.

RESUMO

Ao observar a escrita de um aluno e encontrar um desvio da norma ortográfica do português brasileiro é possível identificar seu nível de conhecimento de ortografia. A partir deste diagnóstico, pode-se desenvolver um tratamento mais personalizado, a fim de melhorar a qualidade de ensino do português brasileiro. Contudo, há poucos recursos para o desenvolvimento da área de processamento de linguagem natural em português brasileiro. Com isso, este trabalho tem o objetivo de viabilizar o desenvolvimento de modelos de aprendizado de máquina que utilizam os desvios ortográficos como objeto de estudo, fornecendo os dados necessários para o treinamento destes modelos. Neste trabalho é criada uma base rotulada de desvios ortográficos do português brasileiro. Os desvios gerados são aqueles mais prováveis de serem encontrados na escrita de um aluno do quinto ano do ensino fundamental, ou seja, são adicionados à base desvios que respeitam a estrutura básica do português brasileiro e não são apenas uma permutação de letras em ordem aleatória. Para sistematizar este conhecimento estrutural da língua foi utilizada uma representação de palavras chamada de etiquetas. Com a manipulação destas etiquetas é possível produzir desvios mais elaborados, que geram palavras possíveis na fonética da linguagem embora sem significado. Por exemplo, o desvio 'porfessor' é considerado válido e adicionado à base, embora 'rpofessor' não seja pois a sequência 'rp' não existe em uma mesma sílaba no português brasileiro. Os desvios gerados são de quatro categorias principais: omissões, substituição não fonológica, substituição fonológica dentro da classe e transposições dentro da sílaba. Um desvio possível da palavra 'professor' para cada categoria, na ordem que foram listadas são 'pofessor', 'profeçor', 'plofessor' e 'porfessor'. Cada desvio da base é detalhado com: sua categoria; os índices referentes à posição na palavra original onde ocorreu a mudança; os caracteres envolvidos no desvio e suas funções dentro da palavra original. Cada desvio também tem como atributo sua palavra original, a separação silábica e a representação fonética de sua origem. Além disso, é fornecida uma interface que gera novas bases rotuladas, originando-se de palavras aleatórias ou escolhidas pelo usuário. Este trabalho visa contribuir para o ensino do português brasileiro fornecendo uma base de dados que possibilita explorar os desvios ortográficos.

Palavras-chave: Processamento de linguagem natural. Modelo de linguagem. Base rotulada. Desvios ortográficos. Português brasileiro.

ABSTRACT

By observing a student's writing and finding a deviation from the Brazilian Portuguese orthographic norm, it is possible to identify his or her level of orthography knowledge. Based on this diagnosis, a more personalized treatment can be developed to improve the quality of teaching Brazilian Portuguese. However, there are few resources available for the development of natural language processing in Brazilian Portuguese. The objective of this work is to enable the development of machine learning models that focus on orthographic deviations, providing the necessary data for training these models. In this study, a labeled database of orthographic deviations in Brazilian Portuguese is created. The deviations generated are those most likely to be found in the writing of a fifth-year elementary school student, that is, deviations that respect the basic structure of Brazilian Portuguese, and are not just a permutation of letters in random order, are added to the base. In order to systematize this structural knowledge of the language, a word representation called 'etiquetas' (tags) is used. By manipulating these tags, it is possible to produce more elaborate deviations that generate phonetically possible but meaningless words. For example, the deviation 'porfessor' is considered valid and added to the database, while 'rpofessor' is not because the sequence 'rp' does not exist within a syllable in Brazilian Portuguese. The generated deviations fall into four main categories: omissions, non-phonological substitution, phonological substitution within the class, and transpositions within the syllable. An example of a possible deviation for each category, in the order listed, for the word 'professor' would be 'pofessor', 'profeçor', 'plofessor' and 'porfessor'. Each deviation in the database is detailed with its category, the indices referring to the position in the original word where the change occurred, the characters involved in the deviation, and their functions within the original word. Each deviation also includes attributes such as the original word, syllabic separation, and the phonetic representation of its origin. Additionally, an interface is provided that generates new labeled databases, originating from random words or words chosen by the user. This work aims to contribute to the teaching of Brazilian Portuguese by providing a database that makes it possible to explore orthographic deviations.

Keywords: Natural Language Processing. Language Model. Labeled Dataset. Orthographic deviations. Brazilian Portuguese.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Este capítulo discorre sobre a motivação para o desenvolvimento deste estudo, a proposta criada para abordar o problema, os desafios encontrados, a contribuição alcançada e por fim a organização do documento.

1.1 MOTIVAÇÃO

No contexto de ensino da língua portuguesa, pouco progresso foi alcançado nas avaliações nacionais ao longo dos últimos anos. De acordo com o Sistema de Avaliação do Ensino Básico, em português 69% dos alunos não chegam ao nível de conhecimento considerado adequado. Essa defasagem de conhecimento em português foi a causa da reprovação de 83,5% dos candidatos a vagas de estágio em 2022, de acordo com o levantamento realizado pelo Nube – Núcleo Brasileiro de Estágios. Portanto, cabe reavaliar o processo de alfabetização da língua portuguesa e definir diretrizes mais claras de ensino, necessidade apontada por MORAIS(2000: 66):

> "... como a ortografia é tratada entre nós mais como tema de verificação que de ensino sistemático, a maioria das escolas do país funciona sem planejar o que espera conseguir na promoção da competência ortográfica de seus alunos a cada série. E como quem não tem metas não antevê aonde quer chegar, não planifica sua ação... pode não conseguir progressos significativos no rendimento que seus alunos expressam ao escrever."

Com o objetivo de melhorar o processo de ensino da língua portuguesa, em (Chacon e Pezarini, 2018) foi criada uma proposta de classificação do desempenho ortográfico infantil. Essa proposta visa identificar o nível de conhecimento de ortografia a partir de um desvio na escrita. Durante este trabalho, chamaremos de “desvio” o que é considerado um erro segundo a norma ortográfica da língua.

Em (Chacon e Pezarini, 2018), são propostas categorias para as diferentes causas de desvios ortográficos, algumas mais graves que representam um nível de alfabetização menor, e outras menos graves, que podem representar apenas o esquecimento de uma convenção. Para elucidar melhor a gravidade dos desvios, considere a palavra “professor”. O desvio ortográfico “pofessor” representa uma omissão de um fonema, enquanto que o desvio “profeçor” representa uma substituição de uma letra por outra, neste caso com o mesmo som. Segundo (Chacon e Pezarini, 2018), esses desvios estão em planos diferentes de conhecimento ortográfico, sendo assim necessário distinguir os autores de tais desvios pois suas necessidades de aprendizado são diferentes. Portanto, tal distinção entre os alunos possibilita um melhor diagnóstico do problema ortográfico, o que auxilia o professor de língua portuguesa a desenvolver um tratamento específico.

1.2 PROPOSTA

Este trabalho se propõe a contribuir para o desenvolvimento de modelos de aprendizado de máquina que usam o português brasileiro como objeto de estudo. A contribuição está na construção de uma base rotulada dos possíveis desvios da norma ortográfica da língua, com foco nos desvios ortográficos mais prováveis de serem encontrados ao observar a escrita de um aluno

em fase de alfabetização (Chacon e Pezarini, 2018). A base gerada possibilita o desenvolvimento de modelos de aprendizado supervisionado, cujo treinamento requer dados rotulados, pois cada desvio é detalhado a fim de facilitar sua codificação.

1.3 DESAFIOS

O desenvolvimento deste trabalho teve como principal desafio filtrar os resultados irrelevantes gerados em cada categoria de desvio ortográfico. Consideramos como irrelevantes os desvios que geram palavras que não respeitam a estrutura básica do português brasileiro, pois o foco deste estudo é gerar desvios prováveis de serem cometidos por alunos do quinto ano do ensino fundamental. Por exemplo, o desvio "porfessor" da palavra "professor" é mais frequente na escrita de alunos que o desvio "prfoessor", já que o aluno com um conhecimento básico de fonética sabe que tal estrutura não é possível no português brasileiro.

1.4 CONTRIBUIÇÃO

Este trabalho traz como contribuição uma base rotulada de desvios ortográficos, uma interface para gerar novas bases e o código utilizado para criar os desvios de cada categoria descrita por (Chacon e Pezarini, 2018). Cada desvio da base é detalhado a fim de que seja possível utilizá-lo como entrada para um modelo de aprendizado de máquina. Além disso, desvios específicos, como por exemplo a troca de 'r' por 'l' gerando 'plofessor', podem ser separados utilizando um filtro nas colunas da base.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento é composto por 5 capítulos. Este capítulo discorre sobre a motivação deste trabalho, a proposta para abordar o problema, as dificuldades encontradas no percurso deste estudo e a contribuição alcançada. O capítulo 2 expõe a base teórica necessária para a manipulação das palavras e geração dos desvios que respeitam a estrutura básica do português brasileiro, assim como a definição das classificações de desvios. O capítulo 3 descreve como as etiquetas foram utilizadas na manipulação das palavras e os métodos aplicados para produzir os desvios de cada categoria. O capítulo 4 explica a implementação em código dos métodos para gerar os desvios. Também apresenta a interface e explica como os desvios são rotulados. O capítulo 5 apresenta as conclusões obtidas neste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentada a base teórica necessária para o desenvolvimento deste trabalho. A primeira seção discorre sobre a importância dos dados para algoritmos de aprendizado de máquina, contribuição feita neste estudo. A segunda seção explica a representação de palavras por etiquetas. Por fim, o capítulo é finalizado com a descrição das categorias de desvios ortográficos sugerida em (Chacon e Pezarini, 2018).

2.1 DADOS EM APRENDIZADO DE MÁQUINA

Em aprendizado de máquina é preciso definir um conjunto de dados de treinamento para a aplicação de algoritmos que podem ser classificados de forma geral em duas categorias, de acordo com o formato de dados fornecido, como aprendizagem supervisionada e não-supervisionada.

A aprendizagem supervisionada utiliza como entrada uma base de dados na forma de pares ordenados (entrada – saída esperada) rotulada para treinar algoritmos, ou seja, é feita a identificação dos tipos de dados para facilitar o reconhecimento dos dados pela máquina. Assim, o algoritmo pode ser treinado com os dados e gerar como saída um conjunto de rótulos pré-definidos como em algoritmos de classificação (resultado categórico), ou gerar um rótulo de saída de um valor real qualquer como nos algoritmos de regressão (resultado numérico). Os dados de entrada podem ser divididos em conjuntos de treinamento e teste, o de treinamento é utilizado para construir o modelo e o de teste para verificar o resultado alcançado.

Diferente do aprendizado supervisionado, no aprendizado não supervisionado não há a rotulagem dos dados. O algoritmo busca as similaridades e padrões entre os dados com algoritmos de transformação e algoritmos de agrupamento (clustering). Os algoritmos de agrupamento usam uma base de critérios já estabelecidos para encontrar padrões que possam dividir os dados por grupos de similaridade, aplicando diversos métodos de agrupamento. Já os algoritmos de transformação têm o objetivo de facilitar a interpretação humana e melhorar desempenho em aprendizagem, e para isso, dado um conjunto de dados original, uma nova representação é criada.

A importância de uma base rotulada está na formação de dados mais significativos e informativos, o que ajuda no reconhecimento e entendimento pelas máquinas que treinarão os modelos através dos algoritmos de aprendizado de máquina. Assim, a rotulagem impacta diretamente no reconhecimento de padrões para uma previsão, se o objetivo é distinguir palavras com desvios ortográficos do português por exemplo, é necessário coletar desvios com atributos que o descrevem, como definido neste trabalho, a posição e caractere da palavra original que foram alterados e como ocorreu esta alteração.

2.2 ETIQUETAGEM

Para gerar os desvios ortográficos que são prováveis de serem cometidos por uma criança, é necessário que a estrutura básica de uma palavra seja traduzida computacionalmente. Com isso, serão gerados desvios mais elaborados, do nível de um aluno que conhece a estrutura da palavra porém a manipula de maneira imprecisa. Por exemplo, a palavra “andando” tem como provável desvio de omissão (quando um fonema é apagado) a construção “andano”, enquanto que o desvio “ndando” é improvável. Isso ocorre pois o aluno sabe que o segundo desvio é impronunciável pois fere a estrutura possível de uma palavra ditada pelo português brasileiro. Para traduzir tal conhecimento estrutural, utilizamos a etiquetagem proposta por (Silva et al.,

2021). As etiquetas são construídas a partir das sílabas, cuja estrutura básica é mostrada na figura 2.1.

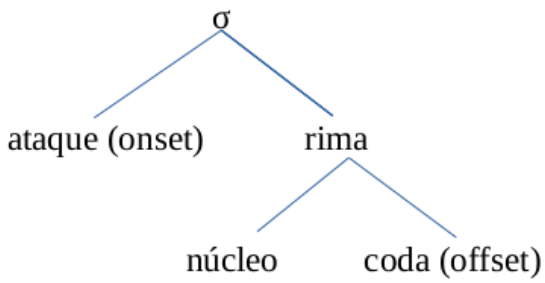


Figura 2.1: Estrutura interna da sílaba. Fonte: (Silva et al., 2021).

Algumas definições são necessárias para a exploração das palavras ao longo deste trabalho: grafema, fonema, e restrições fonotáticas. Grafema, no contexto deste trabalho, pode ser considerado como sinônimo de letra. Fonemas são os sons produzidos pelos falantes e representam as unidades sonoras que formam as palavras de uma língua. Restrições fonotáticas são as possíveis combinações de fonemas numa língua.

Nas sílabas apenas o núcleo é obrigatório, sendo ocupado por uma única vogal ou por duas vogais, casos sendo classificados como núcleo simples e núcleo complexo, respectivamente. Os componentes ataque e coda são opcionais e preenchidos por consoantes. As consoantes de coda formam um subconjunto das consoantes de ataque, devido às restrições de finalização de palavras do português brasileiro. Para as posições de ataque e coda também é possível ocorrer mais de uma consoante, como na sílaba 'pro' da palavra 'professor' e na sílaba 'pers' da palavra perspectiva.

Quanto à representação, Na figura 2.2, SN corresponde ao componente núcleo simples, SA ao ataque simples, SC à coda simples. Os componentes complexos são representados por Cxi, onde x corresponde à classe do fonema (A, C ou N) e i corresponde a posição do fonema em casos complexos, por exemplo na sílaba 'pra', 'p' seria classificado como CA1, por estar na primeira posição do ataque complexo, enquanto que 'r' seria classificado como CA2. As etiquetas O, F, N, e L são subclassificações da classe de consoantes, sendo respectivamente oclusivas, fricativas, nasais e líquidas, de acordo com o modo de articulação. As definições do parágrafo a seguir foram retiradas de (Silva, 2019).

As oclusivas são produzidas por oclusão completa e momentânea no trato vocal, de modo a impedir totalmente a passagem do ar pelo trato, como os sons iniciais das palavras: pato e bola. As fricativas são produzidas quando não há oclusão total do trato, mas uma grande constrição, formada pela aproximação máxima entre dois articuladores, sem que eles se toquem, como nas primeiras consoantes das palavras: fácil e vácuo. As consoantes nasais se o ar é bloqueado na cavidade oral, mas o véu está abaixado, permitindo a propagação do ar pela cavidade nasal, como em: mapa e nata. As consoantes líquidas englobam 3 tipos de consoantes com articulação distinta: taps, laterais e vibrantes. A produção dos taps é semelhante à das vibrantes. A única diferença está no fato de que, nesse caso, há um único período de obstrução à passagem do ar provocado pela batida da ponta da língua na região superior do trato e no qual a voz praticamente desaparece, seguido de um período em que a voz é retomada e o ar passa livremente pelo trato, como em ópera ou prato. A produção das consoantes laterais envolve o bloqueio da corrente do ar em um ponto em torno do centro do trato vocal, com uma oclusão incompleta entre um ou os dois lados da língua e o céu da boca, como por exemplo o início da palavra 'lata'. Por fim, as vibrantes, são produzidas com rápidos períodos de obstrução à

passagem do ar e rápidos períodos em que a obstrução se desfaz, possibilitando a passagem do fluxo de ar. No português brasileiro, os articuladores envolvidos na produção das vibrantes são os lábios, que se juntam e se afastam rapidamente para produzir a vibração.

Como exemplo desta subclassificação, a sílaba "por" será representada por (SAO)(SN)(SCL), pois 'p' é um ataque simples constituído por uma consoante oclusiva, 'o' é um núcleo simples e 'r' é uma coda simples constituída por uma consoante líquida. Esta subclassificação leva em consideração a forma que um som é pronunciado, relevante para avaliar a proximidade sonora entre um desvio e sua palavra original.

Nas etiquetas também é representado o grau do acento, sendo a sílaba tônica representada com 3, pretônica com 1 e postônica com 0.

Variável	**Representação ortográfica**	**Etiqueta**
Consoantes oclusivas	p, b, t, d, c, qu, g, gu	O
Consoantes fricativas	f, v, s, ss, c, x, z, ch, j, g	F
Consoantes nasais	m, n, nh	N
Consoantes líquidas	l, lh, r, rr	L
Vogais	i, e, a, o, u, ẽ, ã, õ	V
Ataque simples	O, F, N ou L	SA
Núcleo simples	V	SN
Coda simples	p, t, d, c, g, f, s, z, m, n, l, r	SC
Primeira posição de ataque ramificado	p, b, t, d, c, g, f, v	CA1
Segunda posição de ataque ramificado	l, r, s, m, n	CA2
Primeira posição de núcleo complexo	i, e, a, o, u, ã, õ	CN1
Segunda posição de núcleo complexo	i, u, e, o	CN2
Primeira posição de coda complexa	n, r	CC1
Segunda posição de coda complexa	s	CC2
Sílaba tônica		3
Sílabas pretônica e postônica		1
Sílaba postônica átona final		0

Figura 2.2: Detalhamento das etiquetas. Fonte: (Silva et al., 2021).

Além disso, as sílabas são agrupadas em colchetes, onde cada grafema é contido por parênteses. Após as sílabas é indicado o grau de acento. Como exemplo a palavra "açúcar" é representada pela etiquetagem: [(SN)]1[(SAF)(SN)]3[(SAO)(SN)(SCL)]0. A primeira sílaba, (SN) corresponde apenas a vogal 'a', tendo como grau de acento 1 já que a sílaba é pretônica. Na segunda sílaba, (SAF) corresponde à consoante 'ç' fricativa (F) do ataque simples (SA), e (SN) corresponde novamente ao núcleo da sílaba 'ú', tendo grau 3 pois é uma sílaba tônica. A última sílaba, com (SAO) representando a consoante 'c' oclusiva (O) do ataque simples (SA), (SN) como núcleo 'a' da sílaba e (SCL) representando uma consoante 'r' líquida (L) da coda silábica (SC). Como a sílaba é postônica, o grau de acento é 0.

Segundo (Silva et al., 2021), as etiquetas traduzem as restrições fonotáticas do português brasileiro, ou seja, uma sequência como <shr> não é permitida em nossa língua, embora seja em outras, como no inglês. Logo, as etiquetas traduzem o conhecimento básico da língua portuguesa que uma criança conhece, equiparando o nível ortográfico do algoritmo com o aluno, sendo portanto, possível reproduzir os desvios deste último.

2.3 CATEGORIZAÇÃO DOS DESVIOS ORTOGRÁFICOS

Nesta seção será explicada a proposta de categorização dos principais desvios ortográficos cometidos por crianças.

Na figura 2.3 são mostradas as diferentes categorias. Cada coluna é formada por diferentes grupos de desvios, que se ramificam nas colunas à direita. Ao longo desta seção será explicado cada grupo de desvios, com exemplos dados em (Chacon e Pezarini, 2018).

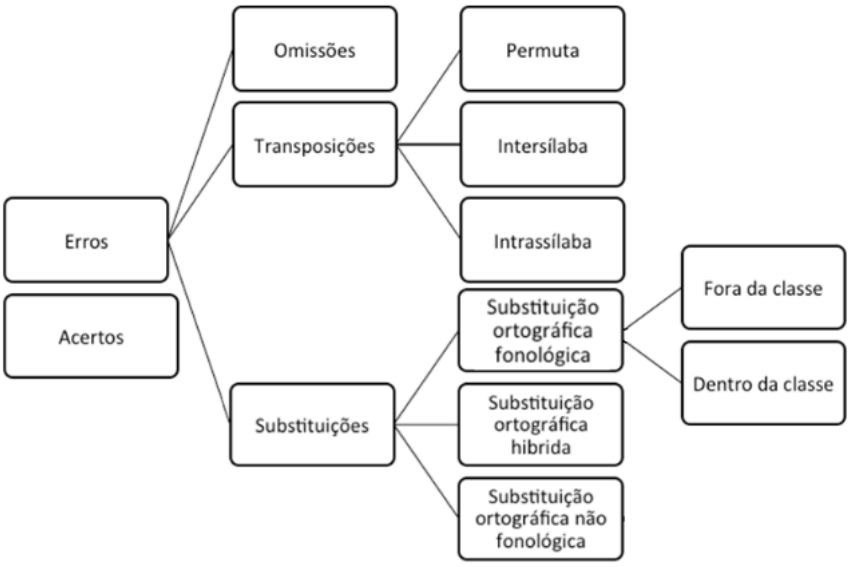


Figura 2.3: Classificação de desvios. Fonte: (Chacon e Pezarini, 2018).

Os desvios são classificados em três principais grupos. As **omissões** quando um fonema é omitido, por exemplo, na palavra "pente" escrita como "pene". As **transposições** quando um fonema é deslocado de sua posição correta, por exemplo, a palavra "professor" escrita como "porfessor". As **substituições** quando um grafema é trocado por outro, por exemplo, a palavra "cebola" escrita como "sebola".

As transposições são ramificadas em três divisões. As **permutas** quando há uma troca de posições de grafemas, como por exemplo, na palavra "serena" registrada como "senera". As **intersílabas** quando há deslocamento de grafema de uma sílaba para outra da mesma palavra, como, por exemplo, na palavra "dentro" registrada como "drento". E, por fim, as **intrassílabas** quando há deslocamento de grafema de uma posição para outra no interior de uma mesma sílaba, como, por exemplo, na palavra "pergunta" registrada como "pregunta".

As substituições também são divididas em outros 3 subgrupos. As **substituições ortográficas fonológicas** quando uma substituição altera o valor fonológico da palavra, como no desvio "calo" da palavra "galo". As **substituições híbridas** quando uma substituição em outros contextos fonológico-ortográficos pode não acarretar mudança fonológica, a palavra "líquido" escrita como "lícido". As **substituições ortográficas não-fonológicas** quando uma substituição não altera o valor fonológico de uma palavra. como a palavra "corta" escrita como "qorta", ou "rato" escrita como "rrato".

A quarta e última coluna da figura 2.3 divide as substituições ortográficas fonológicas em dois grupos. A **substituição fora da classe**, caso o fonema for substituído por outro diferente de sua classe, como "bolacha" escrita como "molacha" (oclusiva substituída por nasal). E a **substituição dentro da classe**, caso o fonema foi substituído por outro de sua mesma classe, como "cola" escrita como "gola", cuja troca ocorreu dentro da classe das oclusivas.

2.4 CONSIDERAÇÕES

A fundamentação teórica apresentada neste capítulo será utilizada ao longo de todo o percurso deste trabalho. A etiquetagem será empregada como um conhecimento estrutural das palavras do português brasileiro para gerar os desvios válidos. A categorização explicada será utilizada para guiar a produção e classificação dos desvios ortográficos. Porém, este trabalho focará na produção de desvios de quatro categorias: omissões, substituições não fonológicas, substituições fonológicas dentro da classe e transposições dentro da sílaba. O fluxograma em 2.4 representa o processo de criação dos desvios de omissão.

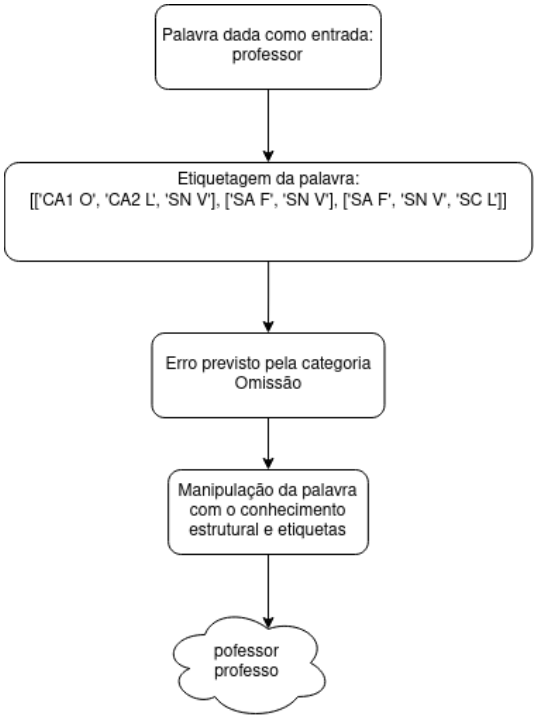


Figura 2.4: Fluxograma da criação de desvios de omissão.

3 MATERIAIS E MÉTODOS

Neste capítulo será explicado como as etiquetas foram utilizadas como conhecimento estrutural do português brasileiro e como foram produzidos os desvios de cada categoria descrita em (Chacon e Pezarini, 2018).

3.1 ETIQUETAS

A etiquetagem das palavras do português brasileiro foi produzida e compactada em um arquivo CSV por (Souza, 2022). O resultado segue este modelo, onde o primeiro valor é a palavra original, o segundo a separação silábica, o terceiro sua representação fonética e por último a etiqueta.

Tabela 3.1: Exemplo de uso das etiquetas

Palavra	Sep. Silábica	Fonética	Etiquetagem
professor	pro.fe.'ssor	pɾo.fe.'sox	[(CA1 O)(CA2 L)(SN V)]1.[(SA F)(SN V)]1.[(SA F)(SN V)(SC L)]3
prol	prol	pɾow	[(CA1 O)(CA2 L)(SN V)(SC L)]3

Para relacionar a etiquetagem com a palavra original a fim de produzir os desvios desejados foi utilizado o índice da string para correlacionar as duas informações, ou seja, o segundo elemento da string "professor" é 'r', logo espera-se que o segundo elemento da lista de etiquetas ['CA O', 'CA L', 'SN V', 'SA F', 'SN V', 'SA F', 'SN V', 'SC L'] seja 'CA L'. Porém, o tamanho da palavra e o tamanho da lista de etiquetas nem sempre são iguais, como no caso de 'professor', em que a palavra é formada por 'ss', cuja representação nas etiquetas é 'SN F'. Para que a correspondência direta entre índices seja possível basta substituir o grafema 'ss' por um caractere especial conhecido pelo algoritmo. Logo, os fonemas compostos serão representados internamente segundo a estrutura 3.1. Portanto, a palavra 'professor' será lida pelo algoritmo como 'profe2or'. Observe que o caso 'gu' foi omitido devido à sua dupla função: em 'água' é um fonema composto, embora em 'pergunta' não seja.

$$dict\_composto = \{qu : 1, ss : 3, ch : 4, nh : 5, lh : 6, rr : 7\} \tag{3.1}$$

Para produzir os desvios relacionados à fonética, esta correspondência direta de índices também é utilizada, porém há uma exceção. A palavra 'profe2or' tem uma correspondência direta com sua representação fonética 'pɾofesox', por exemplo o caracter '2' (traduzido para 'ss') tem o som de 's'. Porém, a palavra 'exceção' tem uma omissão do x em sua representação fonética 'esesãʊ', impossibilitando sua correspondência de índices. Casos de omissão fonética não serão abordados neste trabalho.

3.2 OMISSÕES

Para produzir um desvio ortográfico da categoria de omissões que ainda é pronunciável, basta ocultar os componentes não obrigatórios de uma sílaba, ou seja, os ataques e codas. Nos casos de ataque ou coda complexos, a omissão produzirá seu correspondente simples válido, já que a primeira ou segunda posição de um componente complexo está presente no

conjunto dos possíveis componentes simples. Por exemplo, uma omissão possível para a palavra ‘professor’ é ‘pofessor’, omitindo a consoante líquida ‘r’ em segunda posição de ataque complexo, transformando ‘p’ em um ataque simples também válido.

Na tabela 3.2 são descritas as omissões possíveis para a palavra ‘professor’. Observe que o desvio ‘rofessor’ não está presente, por se tratar de um desvio pouco frequente[1].

Tabela 3.2: Exemplo de omissão

Palavra original	Desvio	Descrição	Mudança
professor	pofessor	Omissão de consoante líquida em posição de ataque complexo	r
professor	professo	Omissão de consoante líquida em posição de coda	r

3.3 SUBSTITUIÇÕES NÃO FONOLÓGICAS

As substituições não fonológicas geram desvios com o mesmo som da palavra original, logo, nesta categoria de desvios é necessário analisar a representação fonética da palavra. Observe que para fazer uma troca fonológica equivalente é necessário checar o som que a letra a ser substituída está realizando na palavra, pois uma fonema pode exercer diferentes sons em diferentes palavras. Por exemplo, a letra ‘c’ representa o som de [s] em cebola, porém som de [k] em casa. Portanto, uma substituição de ‘c’ por ‘k’ seria válida em ‘casa’, embora incoreta em ‘cebola’. Para avaliar essa validade de substituição e produzir desvios com o mesmo som, compara-se a letra a substituir com o valor fonético da letra a ser substituída, se forem iguais a substituição é válida, por exemplo, o desvio ‘kasa’ é válido pois o valor fonético de c é igual a [k], pois a representação fonética de ‘casa’ é ‘kazə’. As substituições válidas são descritas na estrutura de dicionário em 3.2.

$$dictSom = \{qu : k, c : s, ss : s, ç : s, x : s, s : z\} \tag{3.2}$$

Por exemplo, algumas substituições não fonológicas possíveis são descritas em 3.3

Tabela 3.3: Exemplos de substituições não fonológica

Palavra original	Desvio	Descrição	Mudança
professor	profesor	Substituição não fonológica de ss para s	(ss, s)
queijo	keijo	Substituição não fonológica de qu para k	(qu, k)
próximo	prósimo	Substituição não fonológica de x para s	(x, s)

[1]Cf. Adelaide Silva (comunicação pessoal), omissões de consoantes em início de sílaba não são esperadas porque violariam restrições fonotáticas do português brasileiro.

3.4 SUBSTITUIÇÕES FONOLÓGICAS DENTRO DA CLASSE

Os desvios gerados por substituições fonológicas dentro da classe têm o som diferente da palavra original, porém cada fonema é substituído por apenas um outro fonema de sua mesma classe. Observe porém que dependendo da posição do fonema a substituir há diferentes possibilidades, por exemplo, uma oclusiva em posição de ataque simples tem mais possibilidades de substituição comparado com uma oclusiva em posição de coda simples, note na estrutura definida em 3.4.

Tabela 3.4: Etiquetas e seus possíveis ocupantes

Etiqueta	Ocupantes
SC O	p, t, d, c, g
SC F	f, s, z
SC N	m, n
SC L	l, r
SA O	p, b, t, d, c, qu, g
SA F	f, v, s, ss, c, x, z, ch, j, g
SA N	m, n, nh
SA L	l, lh, r, rr
CA2 L	r, l

Também é possível substituir fonemas em posição de ataque complexo. Porém, ressaltando novamente o objetivo deste trabalho de produzir desvios pronunciáveis, é necessário checar se o ataque resultante é válido no português brasileiro. Por exemplo, um possível desvio fonológico, dentro da classe, para “professor” é “plofessor”. Por outro lado, o desvio “alessandla” de “alessandra” não é válido pois o ataque complexo “dl” não existe no português brasileiro. As combinações válidas de ataque e coda complexos são definidas em 3.5.

Tabela 3.5: Ataques, codas e seus ocupantes

Combinação	Ocupantes
Ataque complexo (CA)	pl, pr, ps, pn, bl, br, tl, tr, ts, dr, cl, cr, gl, gr, fl, fr, vl, vr
Coda complexa (CC)	ns, rs
Coda simples (SC)	p, t, d, c, g, f, s, z, m, n, l, r

Todas as combinações obedecem à escala (ou hierarquia) de sonoridade. Em (Ohala e KAWASAKI-FUKUMORI, 1997) é definida a escala de sonoridade da seguinte maneira: oclusiva < fricativa < nasal < líquida < vogal.

A escala se constrói sobre o parâmetro “grau de obstrução à passagem do fluxo de ar no trato vocal”, que é correlacionado à saliência perceptual das consoantes, ou à sua sonoridade. As oclusivas são produzidas necessariamente através de bloqueio total à passagem do fluxo de ar no trato vocal. Por isso, são menos salientes perceptualmente e, portanto, estão no último lugar do ranking de sonoridade. Vogais são produzidas sem oferecer bloqueio à passagem do ar no trato vocal, o que faz delas sons mais salientes em termos perceptuais. Por isso ocupam o topo da escala. Então, as sequências, tanto em ataque como em coda, obedecem a essa escala, o que não implica na ocorrência de todas as combinações bem formadas – como seria ‘dl’, inexistente no português.

Algumas das substituições fonológicas dentro da classe a partir da palavra ‘professor’ são descritas na tabela 3.6.

Tabela 3.6: Exemplos de substituições fonológicas dentro da classe

Palavra original	Desvio	Descrição	Mudança
professor	plofessor	Substituição fonológica dentro da classe líquida	(r, l)
professor	provessor	Substituição fonológica dentro da classe fricativa	(f, v)
professor	profexor	Substituição fonológica dentro da classe fricativa	(ss, x)

3.5 TRANSPOSIÇÕES DENTRO DA SÍLABA

Para gerar transposições dentro da sílaba e obter desvios que respeitem a estrutura do português brasileiro é necessário checar se a transposição produz uma sílaba válida. Por exemplo, a sílaba 'por' tem como desvio válido de transposição a sílaba "pro". Esta validade é checada ao analisar a etiqueta da sílaba original, que neste caso é (SA, SN, SC). Observe que para produzir a sílaba "pro" a etiqueta se transforma em (CA1, CA2, SN), pois 'p' e 'r' são colocados em posição de ataque complexo. Como 'pr' é um dos possíveis ataques complexos da língua portuguesa, como descrito em 3.5, a transposição é válida.

Uma transposição no sentido contrário, ou seja de (CA1, CA2, SN) para (SA, SN, SC), também é possível, como o desvio 'porfessor' de 'professor' e 'defalgrar' de 'deflagrar'. Para checar a validade da transposição nestes casos basta checar se a letra que está em posição CA2 também pode ocupar uma posição de coda simples, SC. Como 'r' e 'l' está no conjunto das possíveis codas simples, descrito em 3.5, as duas transposições citadas anteriormente são válidas. Alguns exemplos válidos de transposições dentro da sílaba são listados na tabela 3.7.

Tabela 3.7: Exemplos de transposições dentro da sílaba

Palavra original	Desvio	Descrição	Mudança
professor	porfessor	Transposição de CA CA SN para SA SN SC	(pro, por)
autografar	autografra	Transposição de SA SN SC para CA CA SN	(far, fra)
adoradoras	adoradoars	Transposição de SA SN SC para SN CC CC	(ras, ars)

3.6 CONSIDERAÇÕES

Neste capítulo foi explicado como as etiquetas foram manipuladas para produzir os diferentes tipos de desvios, porém que ainda mantêm a estrutura básica do português brasileiro válida. No próximo capítulo, o algoritmo utlizado para gerar os desvios será explicado.

4 IMPLEMENTAÇÃO DA SOLUÇÃO PROPOSTA

Neste capítulo serão explorados os pontos chaves do algoritmo desenvolvido em Python para gerar os desvios ortográficos. A primeira seção deste capítulo descreve como as etiquetas de cada palavra são extraídas. O capítulo segue com cada seção explicando a codificação de uma categoria de desvio diferente e finaliza com um manual de uso da interface e uma descrição da base gerada. O código está disponível no repositório `https://github.com/nandozanutto/Gerador_Desvios`.

4.1 EXTRAINDO AS ETIQUETAS

O código abaixo faz a leitura das etiquetas em um dataframe, que será filtrado para encontrar a palavra específica. Os dados retornados deste filtro serão salvos em um dicionário *dict_word* cujas chaves são Palavra, Etiqueta, Sílaba e Fonética.

```
1 import pandas as pd
2 import re
3 df = pd.read_csv('etiquetas.csv')
4 df.drop(df.columns[[4]], axis=1, inplace=True)
5 df.columns = ['Palavra', 'Silaba', 'Fonética', 'Etiqueta']
6 word = "belo:"
7 row_word = df.loc[df['Palavra'] == word]
8 dict_word = row_word.to_dict(orient='records')[0]
```

O código a seguir faz as devidas transformações na etiqueta, para obter duas representações diferentes que serão úteis na manipulação de palavras, *final_etiqueta* e *final_etiqueta2*. A linha 2 transforma a string '[(SA O)(SN V)]3.[(SA L)(SN V)]0' em '['(SA O)(SN V)', '(SA L)(SN V)']', ou seja, extrai as sílabas da palavra e as coloca em uma lista. A linha 8 cria uma representação sem separação de sílabas: '['SA O', 'SN V', 'SA L', 'SN V']'.

```
1 etiqueta = dict_word["Etiqueta"]
2 etiqueta = re.findall(r'\[(.*?)\]', etiqueta)
3 final_etiqueta = []
4 for eti in etiqueta:
5    eti_list = re.split(r'[()]', eti)
6    eti_list = [i for i in eti_list if i]
7    final_etiqueta.append(eti_list)
8 final_etiqueta2 = list(itertools.chain.from_iterable(final_etiqueta))
```

As etiquetas extraídas serão utilizadas nas próximas seções deste capítulo para manipular as palavras mantendo uma estrutura válida do português brasileiro.

4.2 IMPLEMENTAÇÃO DAS OMISSÕES

Para produzir os desvios da categoria de omissões é necessário checar se o grafema a ser omitido não é uma vogal e não é uma consoante que inicia uma sílaba. Isso é feito com "not in nao_emite" na linha 4 do algoritmo abaixo. Os índices possíveis para omissão são salvos em *omissao_list* para serem utilizados no loop da linha 6, onde os desvios são gerados e adicionados na lista *resultList*.

```
1 nao_omite = ['SN V', 'CN1 V', 'CN2 V', 'CA1 O', 'CA1 F', 'CA1 N',\
2 'CA1 L', 'SA O', 'SA F', 'SA N', 'SA L']
```

```
3 for etiqueta in range(len(final_etiqueta2)):
4    if final_etiqueta2[etiqueta] not in nao_omite:
5       omissao_list.append(etiqueta)
6 for omissao in omissao_list:
7    result = palavra[0 : omissao : ] + palavra[omissao + 1 : :]
8    resultList.append(result)
```

4.3 IMPLEMENTAÇÃO DAS SUBSTITUIÇÕES NÃO FONOLÓGICAS

Os desvios da categoria susbtituição não fonológica são produzidos utilizando a representação fonética da palavra, como explicado em 3.3. Logo, o loop externo do algoritmo abaixo itera sobre a sílaba e a representação fonética, tendo como variáveis: *index* o índice da sílaba em sua separação silábica, *silaba* a sílaba analisada na iteração atual e *fonema* a representação fonética da sílaba. O loop interno analisa cada letra da sílaba e seu som correspondente na representação fonética. Portanto a variável *subindex* é o índice da letra em sua sílaba e *som* é o som que a letra da iteração atual exerce.

A estrutura *dict_som* definida em 3.2 é utilizada para encontrar um correspondente de mesmo som de uma letra. Por exemplo, *dictSom.get('c', 'Not found')* retorná 's', que será comparado com o som que a letra 'c' está exercendo na palavra, caso tenha o som de 's' a substituição será executada.

A condicional na linha 7 verifica se o grafema atual é 'ss' (representado internamente por '2') e realiza a substituição por 'ç' caso verdadeiro. A condicional na linha 11 verifica se o grafema atual é 'ss' e se a próxima vogal é 'e' ou 'i', pois 'c' acompanhado dessas vogais tem o mesmo som de 'ss'.

```
1  for (index,silaba), fonema in zip(enumerate(sep_silaba), fonetica):
2     for (subindex, letra), som in zip(enumerate(silaba), fonema):
3        if(dictSom.get(letra, "Not found") == som):
4           substitui(silaba, subindex, index, som, sep_silaba, resultList)
5           lista_sons.append(som)
6           lista_letras.append(letra)
7        if(letra == "2"):
8           substitui(silaba, subindex, index, "ç", sep_silaba, resultList)
9           lista_sons.append("ç")
10          lista_letras.append(letra)
11       if(letra == "2" and silaba[subindex+1] in ["e","i"]):
12          substitui(silaba, subindex, index, "c", sep_silaba, resultList)
13          lista_sons.append("c")
14          lista_letras.append(letra)
```

A função *substitui* troca o grafema na palavra original e salva o desvio em *resultList*. As listas *lista_letras* e *lista_sons* salvam qual grafema foi susbtituído e qual foi a substituição, respectivamente.

4.4 IMPLEMENTAÇÃO DAS SUBSTITUIÇÕES FONOLÓGICAS DENTRO DA CLASSE

Para produzir os desvios da categoria de substituição fonológica dentro da classe é necessário codificar as tabelas 3.4 e 3.5 em uma estrutura de dicionário (onde etiquetas são chaves e ocupantes são valores) e listas, respectivamente.

O loop externo do algoritmo abaixo itera sobre a palavra e sua etiqueta. O loop interno itera sobre todas as possíveis substituições *subs* para a etiqueta. Por exemplo, para a etiqueta 'SC F', as possíveis substituições são 'f', 's' e 'z'.

A condicional da linha 3 checa se a letra substituta é igual a letra original, se sim o desvio não é realizado. A condicional da linha 6 checa se a substituição cria uma coda complexa válida, caso “alessandla” comentado na seção 3.4. As linhas 8 a 11 criam o desvio. As linhas restantes salvam as informações sobre o desvio realizado.

```
1  for (index,letra), etiqueta in zip(enumerate(palavra), final_etiqueta2):
2     for subs in dictEtiquetas.get(etiqueta, []):
3        if(letra == subs):
4           continue #don't replace for the same letter
5        if(etiqueta == "CA2 L"):
6           if(palavra[index-1] + subs not in CA):
7              continue
8        tempList = list(palavra) #word to list
9        tempList[index] = subs #replacing
10       string = "".join(tempList) #list to word
11       resultList.append(string)#saving result
12       lista_letra.append(letra)
13       lista_subs.append(subs)
14       lista_classe.append(etiqueta.split()[1])
```

4.5 IMPLEMENTAÇÃO DAS TRANSPOSIÇÕES DENTRO DA SÍLABA

Os desvios da categoria de transposição dentro da sílaba são produzidos de forma diferente para cada caso de formação da sílaba, como comentado na seção 3.5. Por exemplo, se a sílaba é formada por um ataque complexo e uma vogal simples (condicional da linha 4 do algoritmo abaixo) para produzir um desvio do tipo (SA, SN, SC) é necessário checar se o grafema na posição ‘CA2’ também pode ocupar uma posição de finalização da sílaba, ‘SC’ (condicional da linha 5). Caso essas duas condicionais sejam verdadeiras, a troca das posições 1 e 2 da sílaba é feita[1], como em ‘pro’ para ‘por’.

As condicionais das linhas 11 e 18 realizam o mesmo processo de análise. Note que SC, CA e CC são listas criadas a partir da tabela 3.5.

```
1  for silab_eti, (index_si,si_word) in zip(final_etiqueta, enumerate(sep_si)):
2     caso = [i.split()[0] for i in silab_eti]
3     silab_word_l = list(silab_word)
4     if(caso == ['CA1', 'CA2', 'SN']):
5        if(silab_word_l[1] in SC):
6           troca_result = troca(si_word_l, 1, 2, sep_si, resultList,index_si)
7           lista_desc.append("Transposição de CA CA SN para SA SN SC")
8           lista_troca.append(troca_result)
9           lista_silab.append(silab_word)
10
11    if(caso == ['SA', 'SN', 'SC']):
12       if((silab_word_l[0] + silab_word_l[2]) in CA):
13          troca_result = troca(si_word_l, 1, 2, sep_si, resultList,index_si)
14          lista_desc.append("Transposição de SA SN SC para CA CA SN")
15          lista_troca.append(troca_result)
16          lista_silab.append(silab_word)
17
18       if((silab_word_l[0] + silab_word_l[2]) in CC):
19          troca_result = troca(si_word_l, 0, 1, sep_si, resultList,index_si)
20          lista_desc.append("Transposição de SA SN SC para SN CC CC")
21          lista_troca.append(troca_result)
```

[1]Note que os índices da sílaba começam em zero.

```
22          lista_silab.append(silab_word)
```

4.6 INTERFACE

A interface criada para gerar bases de desvios é apresentada em 4.1. Os campos de 1 a 4 recebem a quantidade de desvios desejados de cada categoria. O campo 6 recebe uma lista de palavras, separadas por espaço, que serão utilizadas para gerar os desvios. Ao acionar o botão 5 os desvios serão gerados e mostrados em "Informações ao Usuário", campo 8. Caso o campo 7 esteja acionado, ao clicar em 5, será gerado um arquivo excel com os desvios criados e seus atributos.

Figura 4.1: Interface do gerador.

Há duas formas de utilizar a interface. A primeira é inserir uma lista de palavras em 6. Neste caso, ao clicar em 5, os desvios gerados serão todos os possíveis para as palavras inseridas, ou seja, os campos de categorias (1 a 4) são ignorados pelo gerador. A segunda forma é inserindo a quantidade desejada para cada categoria de desvio nos campos de 1 a 4, que gerará desvios para palavras aleatórias. A figura 4.2 exemplifica esta última forma de utilizar a interface.

4.7 BASE CRIADA

Como exemplo de base, em 4.3 é apresentada a base gerada a partir de um uso da interface como em 4.2. Observe que a coluna de índice varia de acordo com a categoria de desvio. Em omissões e substituições fonológicas, o índice é referente a qual posição da palavra ocorreu a mudança do caractere. Na categoria de substituição não fonológica, o primeiro índice é referente à posição da sílaba na palavra e o segundo índice referente à posição do caractere dentro da sílaba. Em transposições a coluna de índice é formada pela posição da sílaba na palavra

Figura 4.2: Exemplo de uso da interface.

e as posições dos dois caracteres dentro da sílaba que foram transpostos. A coluna mudança descreve o que foi alterado para gerar os desvios, por exemplo no desvio da linha 3 em 4.3, 'ç' foi trocado por 's', logo a mudança é igual a ('ç','s'). A coluna de descrição fornece detalhes sobre o desvio, por exemplo, 'SC F' na linha 2 descreve que foi omitido um caractere da classe das fricativas e que estava em posição de coda simples.

	Palavra Original	Separação Silábica	Fonética	Desvio	Categoria de Desvio	Índice	Mudança	Descrição
2	carismaticos	ca.ris.'ma.ti.cos	ka.riz.'ma.ʧi.kʊs	carimaticos	Omissão	(4, '-')	('s', '-')	SC F
3	gravaçao	gra.va.'çao	gra.va.'sɐ̃ʊ̃	gravasao	Substituição não fonológica	(2, 0)	('ç', 's')	SA F
4	capeto	ca.'pe.to	ka.'pe.tʊ	papeto	Substituição fonológica	(0, '-')	('c', 'p')	SA O
5	esferograficas	es.fe.ro.'gra.fi.cas	es.fe.ro.'gra.fi.kɐs	esferogarficas	Transposições	(3, 1, 2)	('gra', 'gar')	CA CA SN - SA SN SC

Figura 4.3: Exemplo de base gerada.

4.8 CONSIDERAÇÕES

Neste capítulo foram apresentados os aspectos práticos deste estudo. Foram definidos os códigos para gerar os desvios de cada categoria e explicado como as regras e exceções derivadas da estrutura do português brasileiro foram codificadas. Descreveu-se a interface e como utilizá-la para gerar novas bases. Finalmente, os atributos que acompanham os desvios na base são esclarecidos.

5 CONCLUSÃO

Este trabalho foi desenvolvido a fim de gerar ferramentas que viabilizem o avanço de pesquisas na área de processamento de linguagem natural do português brasileiro, especificamente nos desvios ortográficos da norma culta da linguagem. Objetivando esta contribuição, foi proposta uma base de desvios ortográficos , onde cada desvio é detalhado com atributos que permitem sua utilização em modelos de aprendizado de máquina. Também foi criada uma interface para gerar novas bases rotuladas, com desvios a partir de palavras aleatórias ou escolhidas pelo usuário.

Desvios ortográficos comuns na escrita de um aluno podem ser estudados de forma mais profunda com a base proposta neste estudo, sendo possível melhor diagnosticar as dificuldades no aprendizado de um aluno e desenvolver tratamentos personalizados. Como sequência deste estudo, propõe-se complementar a geração de desvios ortográficos com os casos que foram desatendidos neste trabalho. Como exemplo, os casos de omissão fonética, onde a representação fonética das palavras foge do padrão, e as categorias de desvios não trabalhadas, como as transposições intersílaba.

REFERÊNCIAS

Chacon, L. e Pezarini, I. O. (2018). Gradiência na correspondência fonema/grafema: uma proposta de caracterização do desempenho ortográfico infantil. Em Editorial, P., editor, *Tópicos em Transtornos de Aprendizagem Parte VI*. Booktoy, Ribeirão Preto.

MORAIS, A. G. (2000). *Ortografia: ensinar e aprender*. Ed. Ática, São Paulo.

Ohala, J. e KAWASAKI-FUKUMORI, H. (1997). Alternatives to the sonority hierarchy for explaining segmental sequential constraints. Em Eliasson, S. e Jahr, E. H., editores, *Language and its ecology - Essays in memory of Einar Haugen*, Berlin. De Gruyter Mouton.

Silva, A. H. (2019). *Língua Portuguesa I: Fonética e Fonologia*. IESDE BRASIL S/A.

Silva, A. H., Silva, F., Carvalho, W. e Chacon, L. (2021). A generic representation for orthographic structure in texts written by children. Em 7^{rd} *International Meeting on Speech Sciences*, páginas 98–103, Minas Gerais.

Souza, L. V. D. (2022). Uma comparação entre a influência de diferentes representações silábicas sobre a detecção de semelhança entre palavras em modelos pln.