UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INSTITUTO DE COMPUTAÇÃO

CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CRISTIAN DIAMANTARAS VILELA
GABRIEL RODRIGUES CUNHA

COLETA E ARMAZENAMENTO DE DADOS MORFOLÓGICOS NA LÍNGUA PORTUGUESA

RIO DE JANEIRO

2024

CRISTIAN DIAMANTARAS VILELA
GABRIEL RODRIGUES CUNHA

COLETA E ARMAZENAMENTO DE DADOS MORFOLÓGICOS NA LÍNGUA PORTUGUESA

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. João Carlos Pereira Da Silva

Co-orientadora: Profa. Daniela Cid de Garcia

RIO DE JANEIRO

2024

Vilela, Cristian Diamantaras
V699c Coleta e armazenamento de dados morfológicos na língua portuguesa / Cristian Diamantaras Vilela e Gabriel Rodrigues Cunha. – 2024.

69 f.

Orientadora: João Carlos Pereira da Silva.
Coorientadora: Daniela Cid de Garcia.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)- Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2024.

1. Processamento de imagem natural. 2. Geração de corpus. 3. Morfologia. 4. Língua portuguesa. I. Cunha, Gabriel Rodrigues. II. Silva, João Carlos Pereira da (Orient.). III. Garcia, Daniela Cid de (Coorient.). IV. Universidade Federal do Rio de Janeiro, Instituto de Computação. V. Título.

CRISTIAN DIAMANTARAS VILELA
GABRIEL RODRIGUES CUNHA

COLETA E ARMAZENAMENTO DE DADOS MORFOLÓGICOS NA LÍNGUA PORTUGUESA

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 07 de agosto de 2024

BANCA EXAMINADORA:

Documento assinado digitalmente
gov.br JOAO CARLOS PEREIRA DA SILVA
Data: 16/08/2024 19:07:02-0300
Verifique em https://validar.iti.gov.br

_____
Prof. João Carlos P. da Silva, D.Sc. (UFRJ)

Documento assinado digitalmente
gov.br DANIELA CID DE GARCIA
Data: 19/08/2024 14:15:06-0300
Verifique em https://validar.iti.gov.br

_____
Profa. Daniela Cid de Garcia, D.Sc. (UFRJ)

Documento assinado digitalmente
gov.br SILVANA ROSSETTO
Data: 17/08/2024 07:22:59-0300
Verifique em https://validar.iti.gov.br

_____
Profa. Silvana Rossetto, D.Sc. (UFRJ)

Documento assinado digitalmente
gov.br SABRINA LOPES DOS SANTOS
Data: 19/08/2024 12:28:49-0300
Verifique em https://validar.iti.gov.br

_____
Profa. Sabrina Lopes dos Santos, D.Sc. (UFRJ)

AGRADECIMENTOS

Gostaríamos de agradecer aos nossos orientadores, o professor João Carlos Pereira Da Silva e a professora Daniela Cid de Garcia, por todo o seu suporte durante a elaboração desse trabalho. A disposição e solicitude demonstrada por ambos foi essencial durante toda essa jornada. Sem as suas orientações precisas este trabalho não seria possível.

Expressamos nossos reconhecimentos à toda a comunidade acadêmica da Universidade Federal do Rio de Janeiro (UFRJ), da qual tivemos o privilégio de participar. Agradecemos todos os professores, colegas e funcionários que, direta ou indiretamente contribuíram para o nosso desenvolvimento pessoal e intelectual.

A linguagem é um processo de livre criação; suas leis e princípios são fixados, mas a maneira como os princípios de geração são usados é livre e infinitamente variada. Mesmo as palavras e frases mais simples podem ser construídas de maneira nova e altamente complexa e intricada.

Noam Chomsky

RESUMO

Este trabalho teve como objetivo criar um processo estruturado de coleta de informações morfológicas da língua portuguesa para um banco de dados com o intuito de facilitar análises e estudos sobre o tema. O processo desenvolvido é composto por três módulos independentes. O primeiro módulo é responsável pela geração de corpus, onde arquivos PDF ou de imagem são convertidos em arquivos de texto. O segundo módulo realiza o processamento desses textos, extraindo informações morfológicas e estruturando-as em um banco de dados. Por último, o terceiro módulo realiza análises sobre esses dados, respondendo perguntas específicas referentes a palavras, sufixos e classes gramaticais. Foram utilizados três corpora nesse processo: um de notícias do setor elétrico e dois de autoria própria, sendo o primeiro composto por livros infantis e o segundo por cordéis. Os resultados mostraram que a ferramenta criada gerou informações coerentes em relação as perguntas, o que pode ser corroborado pela Lei de Zipf, que define um comportamento comum em linguagens naturais, demonstrando que o processo desenvolvido é eficaz para a coleta e análise de informações morfológicas da língua portuguesa.

Palavras-chave: processamento de linguagem natural; gramática; morfologia; análise morfológica; sufixo; Python; banco de dados; língua portuguesa; corpus.

ABSTRACT

This paper aimed to create a structured process for collecting morphological information of the Portuguese language into a database to facilitate analyses and studies on the subject. The developed process is composed of three independent modules. The first module is responsible for corpus generation, where PDF or image files are converted into text files. The second module processes these texts, extracting morphological information and structuring it into a database. Finally, the third module performs analyses on this data, answering specific questions regarding words, suffixes, and grammatical classes. Three corpora were used in this process: one of news from the electric sector and two of our own creation, the first composed of children's books and the second of cordel literature. The results showed that the created tool generated consistent information concerning the questions, which can be corroborated by Zipf's Law, which defines a common behavior in natural languages, demonstrating that the developed process is effective for collecting and analyzing morphological information of the Portuguese language.

Keywords: natural language processing; grammar; morphology; morphological analysis; suffix; Python; database; Portuguese language; corpus.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

PLN	Natural Language Processing
NLTK	Natural Language Toolkit
SGBDR	Sistema de Gerenciamento de Banco de Dados Relacional
IFE	Informativo Eletrônico do Setor Elétrico
GESEL	Grupo de Estudos do Setor Elétrico
UFRJ	Universidade Federal do Rio de Janeiro
ER	Entidade Relacionamento

LISTA DE SÍMBOLOS

GiB	GibiByte, equivalente a 2^{30} bytes
[X]	Variável que representa uma palavra qualquer

SUMÁRIO

1 INTRODUÇÃO

A língua portuguesa, com sua rica história e complexa estrutura, fascina e desafia estudiosos. Compreender os mecanismos que regem a formação de palavras e frases é fundamental para entendermos como nos comunicamos, aprendemos e processamos informações. Nesse contexto, o presente trabalho propõe a criação de uma ferramenta para facilitar o estudo da morfologia a fim de auxiliar pesquisadores da área do aprendizado linguístico.

Como expressado por (RASTLE, 2019), a morfologia é um componente essencial no desenvolvimento da leitura, destacando que a análise morfológica contribui significativamente para a compreensão e reconhecimento de palavras. Com base nessas informações, ficou evidente a necessidade de criar ferramentas que pudessem apoiar a análise morfológica de textos em português. O que poderia facilitar, além da análise e do estudo, a seleção e produção de material em salas de aula.

A proposta inicial era de criarmos uma ferramenta classificadora de livros infantis baseada na complexidade do seu texto a partir de dados morfológicos, com o intuito de facilitar a escolha dos livros propostos por faixa etária. Já existe um sistema de classificação etária para diversas mídias no Brasil, o Classind[1], mas não existe um sistema oficial para livros. A ideia seria avaliar analiticamente se um livro é apropriado ou não para uma determinada idade. Porém, não foi possível seguir por este caminho, visto que não havia dados suficientes para treinar um modelo de classificação. Por conta disso, foi necessário coletar e criar uma base de dados morfológicos dos textos, o que permitiria a criação da ferramenta de classificação em um momento futuro.

O objetivo principal deste trabalho é criar um repositório de dados morfológicos da língua portuguesa para facilitar análises e estudos que dependam destes dados. A motivação para realizar o pré-processamento reside na ausência de uma fonte única de dados, tornando necessário padronizar o formato. Além disso, a etapa de separação do sufixo das palavras é um processo complexo e inexato, exigindo um tratamento específico. O que se pretende extrair dos corpora são informações sobre a palavra, sua classe gramatical, sufixo, corpus ao qual pertence

[1] https://www.gov.br/mj/pt-br/assuntos/seus-direitos/classificacao-1

e a frase dentro do corpus onde a palavra está inserida. Essas informações serão aplicadas no módulo de análises para responder a perguntas específicas, que serão descritas no Capítulo 5.

Esse trabalho é composto por três funcionalidades principais que podem ser separadas em módulos distintos: Módulo de Geração de Corpus, que a partir de um arquivo de texto em formato *portable document format* (pdf) ou um arquivo de imagem, gera um arquivo de texto simples contendo o texto do arquivo de entrada; Módulo de Processamento, que a partir de um arquivo simples cria uma base de dados contendo as palavras, sufixos, classes gramaticais e o corpus ao qual pertencem; e o Módulo de Análises que, a partir da consulta à base de dados, gera análises que respondem a algumas perguntas como ocorrência de palavras, de sufixos, palavras por classe gramatical, entre outras. Essas perguntas foram elaboradas em conjunto com a Professora Doutora Daniela Cid, uma especialista no tema, garantindo que as análises fossem alinhadas com as necessidades da área de estudo.

Para entender o funcionamento morfológico das palavras, incluindo a formação de frases, palavras, radicais e sufixos, utilizamos bibliotecas de processamento de linguagem natural em Python, como NLTK[2] e Spacy[3]. Como são necessários dados textuais para processamento de linguagem natural, foram buscados corpora que tivessem uma grande quantidade de frases e palavras. A partir disso, foram utilizados três corpora distintos no projeto, um já existente e dois novos, que foram criados a partir do módulo de geração de corpus, sendo um de livros infantis e outro de cordéis.

Juntando os conhecimentos linguísticos e computacionais, foi possível modelar e criar um processo que extraísse todas estas informações morfológicas de arquivos no formato de texto. Além da criação do processo, foi criado o banco de dados para armazenar os resultados obtidos. O banco de dados tem um papel fundamental, pois, é a partir dele que toda a etapa de estudo e avaliação dos resultados é possível.

No próximo capítulo, abordaremos os conceitos da morfologia da língua portuguesa, destacando a importância da formação de frases, palavras, radicais e sufixos, explicando conceitos que serão importantes para interpretar os resultados. Em seguida, exploraremos os métodos de processamento de linguagem natural utilizados para a extração de dados morfológicos de textos em português, detalhando o uso das bibliotecas Python NLTK e Spacy, e a criação do banco de dados para armazenar e avaliar os resultados obtidos.

[2] https://www.nltk.org/

[3] https://spacy.io/

2 CONCEITUALIZAÇÃO: GRAMÁTICA DA LÍNGUA PORTUGUESA

Este capítulo é destinado à explicação de todos os conceitos que são importantes no decorrer da leitura desta monografia. Os principais conceitos da língua portuguesa utilizados na elaboração deste trabalho foram: Palavras, Radicais, Sufixos e Classes Gramaticais, entretanto outras nomenclaturas e conceitos também serão abordados a seguir. Os prefíxos, apesar de serem relevantes na morfologia, não serão contemplados neste trabalho, pois durante a definição dos objetos de estudo foi decidido dar ênfase nos sufixos, afim evitar aumentar a complexidade do projeto.

2.1 FRASES, PALAVRAS, FONEMAS E MORFEMAS

Na língua portuguesa escrita, podemos definir *palavra* como sendo uma sequência qualquer de caracteres que ocorra entre espaços e/ou sinais de pontuação (BASÍLIO, 1987). Elas normalmente são construídas a partir de letras e podem ser agrupadas em diferentes classes gramaticais, como substantivos, verbos, adjetivos, advérbios, preposições, conjunções e interjeições.

A linguagem é uma complexa construção que possui várias camadas de significado e estrutura. Em seu nível mais macro, encontramos as frases, que são escritos que apresentam um sentido completo, podendo conter apenas uma ou várias palavras (ABAURRE; PONTARA, 2006).

Morfema pode ser definido como a menor unidade linguística portadora de significado. (GONÇALVES, 2019). Eles podem ser palavras inteiras, como “gato”, ou partes de palavras, como os afixos “anti-” ou “re-”, que modificam o significado da palavra à qual são adicionados.

Existem dois tipos principais de morfemas, os livres e os presos. Os livres são aqueles que podem constituir palavras independentes, como é o caso da palavra “papel”. Já os presos são aqueles que não possuem um significado próprio e precisam se ligar a outras palavras, que é o caso do sufixo “-rão” (DE QUADROS; STUMPF; LEITE, 2013). Morfemas presos também podem ser afixos, como os prefixos e sufixos:

- Morfemas prefixos: “pré-” em “pré-história”
- Morfemas sufixos: “-ção” em “navegação”

Os morfemas também podem ser classificados em lexicais e gramaticais, sendo que os lexicais têm significação externa, referindo-se a elementos do mundo real, enquanto morfemas gramaticais estão relacionados às categorias e relações gramaticais como artigos, preposições, conjunções, entre outros. Essa análise morfológica é essencial quando queremos observar comportamentos repetidos ou padronizados em linguagens.

2.2 A ESTRUTURA DE UMA PALAVRA

A estrutura das palavras é organizada em torno do que tradicionalmente é chamado de "radical" ou "raiz" (CUNHA; CINTRA, 2016). O radical é responsável por unir as palavras de uma mesma família e transmitir uma base comum de significação. Ao radical são adicionados os morfemas gramaticais, que podem ser desinências, um afixo ou uma vogal temática. Vale lembrar que desinências também são conhecidas como morfema flexional e afixos como morfema derivacional. As desinências desempenham um papel crucial ao indicar o gênero e número em substantivos, adjetivos e alguns pronomes, assim como o número e pessoa em verbos. Abaixo alguns exemplos destes morfemas:

- Morfemas derivacionais:
    - "-ção" (nominalização: mudança de classe gramatical de uma palavra)
        - Verbo: Informar
        - Derivado: Informação
    - "re-" (prefixo que indica repetição ou retorno)
        - Verbo: Fazer
        - Derivado: Refazer
    - "-ista" (formação de substantivo)
        - Base: Jornal
        - Derivado: Jornalista
- Morfemas flexionais:
    - "-s" (plural em substantivos)
        - Base: Cadeira
        - Plural: Cadeiras
    - "-ou" (pretérito perfeito do indicativo em verbos)
        - Infinitivo: Falar

- Pretérito Perfeito: Falou

2.3 AFIXOS

Os afixos ou morfemas derivacionais são elementos que geralmente modificam de maneira específica o sentido do radical ao qual são adicionados. Existem dois tipos principais de afixos: os prefixos, que se posicionam antes do radical, e os sufixos, que se encontram após o radical. Apesar de o prefixo ser importante para a estruturação das palavras, neste trabalho, focaremos somente no processo de derivação por sufixação.

Os sufixos, ao contrário das desinências, que se limitam a indicar gênero, número ou pessoa sem alterar o sentido lexical, têm o poder de transformar substancialmente o radical.

- Exemplos:
  - Na palavra “terroso”, o sufixo “-oso” transforma o substantivo “terra” em um adjetivo;
  - Em “terreiro”, o sufixo “-eiro” converte o substantivo “terra” em outro substantivo;
  - Em “novinho”, “-inho” modifica o adjetivo “novo” para formar um diminutivo;
  - “novamente”, o sufixo “-mente” transforma o feminino do adjetivo “novo” em um advérbio.

A distinção tradicional entre sufixos e desinências, de certa maneira, pode ser simplificada considerando apenas seu aspecto visual e fonético. Alternativamente, pode-se distinguir esses morfemas pelo seu aspecto funcional, classificando as desinências como morfemas flexionais e os sufixos como morfemas derivacionais. Nesse caso, as características de tempo, modo e as formas nominais do verbo seriam incorporadas às desinências.

2.4 PROCESSO DE FORMAÇÃO DE PALAVRAS

A formação de palavras é um processo morfossintático sujeito a debates entre linguistas contemporâneos quanto à sua inclusão na morfologia, léxico, semântica ou sintaxe (CUNHA; CINTRA, 2016). Alguns processos morfossintáticos possibilitam a criação de novas unidades com base em morfemas lexicais, utilizando a derivação por sufixos e procedimentos de composição.

Um componente essencial da formação de palavras é a derivação sufixal, que é responsável pela geração de substantivos, adjetivos, verbos e até advérbios. Os sufixos são classificados como nominais quando agregados a radicais para originar substantivos ou adjetivos, verbais quando ligados a radicais que geram verbos, e adverbiais, exemplificado pelo sufixo -mente, adicionado à forma feminina de um adjetivo.

Dentre os sufixos nominais, destacam-se os aumentativos e diminutivos, que conferem um valor mais subjetivo do que lógico. Alguns exemplos são: -ão, -anzil, -aréu, -arra, -eirão, entre outros. A formação de verbos, por sua vez, ocorre pela adição de terminações como -ar, -er e -ir a substantivos e adjetivos, como em “esquiar”, “radiografar”, “vender” e “adoçar”. No âmbito adverbial, o sufixo único em português é -mente, derivado do substantivo latino mens, mentis, inicialmente associado à intenção e posteriormente à maneira. As palavras “bondosamente”, “nervosamente” e “fracamente” são exemplos deste caso.

Em resumo, a formação de palavras envolve processos morfossintáticos que incorporam a derivação sufixal para criar unidades lexicais. A variedade de sufixos e suas aplicações reflete a riqueza e complexidade do sistema morfológico da língua portuguesa.

Como esta seção está reservada para conceitualização dos aspectos morfológicos de formação das palavras no português, cabe aqui declararmos uma convenção que tomamos para definir afixos que parecem sufixos, mas que não são de fato sufixos. Estes serão descritos ao longo deste trabalho como pseudosufixos. Um exemplo de pseudosufixo é dado na palavra “guitarra”, onde há a possibilidade de separarmos o radical “guita” e o sufixo -arra mesmo a palavra “guita” sendo uma palavra com significado diferente e não sendo um radical para a palavra “guitarra”. Na Tabela 1 é mostrada a lista dos principais sufixos que foram utilizadas neste trabalho. No Apêndice A está contida a lista completa de sufixos, contendo as flexões de gênero e o plural dos sufixos nominais.

Tabela 1: Lista de Sufixos utilizados

CLASSE	SUFIXOS
Nominal / Noun	‘acho’, ‘aco’, ‘aco’, ‘aço’, ‘ada’, ‘ada’, ‘ado’, ‘agem’, ‘agem’, ‘aico’, ‘al’, ‘alhão’, ‘ama’, ‘ame’, ‘ança’, ‘ano’, ‘ão’, ‘ão’, ‘aréu’, ‘aria’, ‘aria’, ‘aria’, ‘ário’, ‘arra’, ‘ata’, ‘ato’, ‘ável’, ‘az’, ‘ção’, ‘dade’, ‘dor’, ‘douro’, ‘eco’, ‘edo’, ‘eira’, ‘eirão’, ‘eiro’, ‘eiro’, ‘ela’, ‘ela’, ‘ena’, ‘ença’, ‘ência’, ‘ência’, ‘enho’, ‘enho’, ‘eno’, ‘ense’, ‘ento’, ‘eo’, ‘eria’, ‘ês’, ‘eu’, ‘ez’, ‘eza’, ‘ia’, ‘iaco’, ‘ica’, ‘ice’, ‘icho’, ‘ície’, ‘iço’, ‘inho’, ‘ino’, ‘into’, ‘isco’, ‘ismo’, ‘ismo’, ‘ista’, ‘ista’, ‘ível’, ‘ivo’, ‘lento’, ‘mento’, ‘nte’, ‘onho’, ‘or’, ‘or’, ‘oso’, ‘ote’, ‘óvel’, ‘rio’, ‘sor’, ‘tério’, ‘tica’, ‘tor’, ‘tório’, ‘uça’, ‘ude’, ‘udo’, ‘ume’, ‘ura’, ‘ura’, ‘úvel’, ‘zarrão’, ‘zinho’
Verbal / Verb	‘ar’, ‘ear’, ‘ear’, ‘ecer’, ‘er’, ‘escer’, ‘icar’, ‘ir’, ‘iscar’, ‘itar’
Adverbial / Adv	‘mente’

3 CONCEITUALIZAÇÃO: PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural (PLN) é um campo da inteligência artificial que se concentra na interação entre computadores e a linguagem humana (LIU *et al.*, 2022). Entre as suas finalidades, estão o desenvolvimento de algoritmos e modelos de aprendizado de máquina capazes de gerar texto de maneira semelhante a um ser humano, visando a capacidade de compreender e interpretar linguagem humana. As aplicações de PLN abrangem diversas vertentes, desde a tradução automática de textos (ZONG; HONG, 2018), até *chatbots* (RAMADITIYA *et al.*, 2021).

Nesse estudo, o processamento de linguagem natural foi feito utilizando a linguagem de programação Python com o auxílio das bibliotecas NLTK e Spacy. O primeiro, no entanto, possui funcionalidades limitadas para o processamento de textos na língua portuguesa.

O Spacy fornece uma seleção maior de operações para a língua portuguesa. O PLN do Spacy é capaz de receber um texto em linguagem natural, partir esse texto em frases, tokenizar as frases em vocábulos e sinais de pontuação. Tokenização é a tarefa de separar textos em segmentos menores chamados tokens. Tokens podem ser palavras, sinais de pontuação, números, e outros elementos que podem compor uma frase. Por exemplo, a frase “Fui ao mercado, estava muito cheio.” é composta pelos seguintes tokens: ‘Fui’, ‘ao’, ‘mercado’, ‘,’, ‘estava’, ‘muito’, ‘cheio’, ‘.’. Tokens também podem conter mais de uma palavra, como é o caso de ‘guarda-chuva’ e ‘dente-de-leão’. O processo de tokenização auxilia na compreensão de contexto e no desenvolvimento de modelos de PLN. Esse processo é utilizado como base de outros algoritmos de pré-processamento, como o POS tagging, que será explicado mais à frente. Seguindo as normas da forma padrão da língua portuguesa, é possível realizar essa etapa seguindo regras computacionalmente simples. Por exemplo, palavras são separadas entre si por um espaço, pontuação como reticências (...) vírgulas (,) e ponto e vírgulas (;) devem seguir o final de uma palavra.

De acordo com (FRAKES; FOX, 2003), *stemmers* são utilizados para agrupar palavras com sintaxe semelhante. A técnica de *stemming* mais utilizada é a remoção de afixos, produzindo uma forma raiz da palavra chamada de “stem” que aproxima o morfema raiz de uma palavra. É a parte que denota o sentido, ou ideia, principal de uma palavra. Por exemplo, considerando a palavra “desconectado”, removendo o prefixo des- e o sufixo -ado obtém-se a raiz “conect”. Os termos “reconectado”, “conectou” e “reconectará” também compartilham essa mesma raiz.

Embora o *stemming* seja fundamental para algumas tarefas de PLN, sua implementação pode se mostrar bastante complexa. Uma das maiores armadilhas na língua portuguesa para qualquer algoritmo desse tipo é a existência de afixos falsos, que podem ser pseudoprefixos ou pseudosufixos. Considerando uma palavra como “deste”, um sujeito em fase de alfabetização pode erroneamente confundir as três primeiras letras da palavra como o sufixo indicador de negação des-, o que indicaria um pseudoprefixo. Uma instância de pseudosufixo pode ser encontrada no final da palavra “Janeiro”, que pode ser erroneamente apontado como o sufixo indicador de profissão -eiro.

Um token contido dentro de uma frase cumpre um papel dentro desse corpo semântico. De acordo com (CHICHE; YITAGESU, 2022), *part-of-speech tagging* (POS tagging) é o processo de atribuir automaticamente os rótulos às palavras de uma frase, baseado no papel que a palavra cumpre dentro da frase. Uma POS tag é uma classificação gramatical que comumente inclui verbos, adjetivos, advérbios, substantivos, sinais de pontuação, números etc. POS tagging é um método bastante utilizado em diversas aplicações de processamento de linguagem natural.

Em alguns casos, é possível determinar o *POS tag* de um termo considerando apenas o próprio termo. Como é o caso da palavra “menina”, que é um substantivo feminino independente do contexto em que ela estiver inserida. Porém, em alguns casos essa distinção não é tão simples. A palavra “casarão”, por exemplo, pode ser um substantivo no aumentativo da palavra “casa” ou pode ser uma conjugação do verbo “casar”. Nesse caso, é impossível distinguir um do outro considerando o termo individual. Para resolver essa ambiguidade, é considerada a posição desse termo dentro da estrutura da frase. No caso da oração “eles se casarão no casarão”, é evidente que a primeira ocorrência do termo “casarão” consiste de uma conjugação do verbo “casar”, já a segunda ocorrência, que é precedido pelo pronome “no”, se refere a um local, mais precisamente, a uma casa. Utilizando, na análise da estrutura de uma frase, regras semelhantes ao exemplo citado, é possível determinar a função de cada um dos seus termos. Na Tabela 2 foram listadas as POS tags presentes no modelo da língua portuguesa do Spacy e os seus significados.

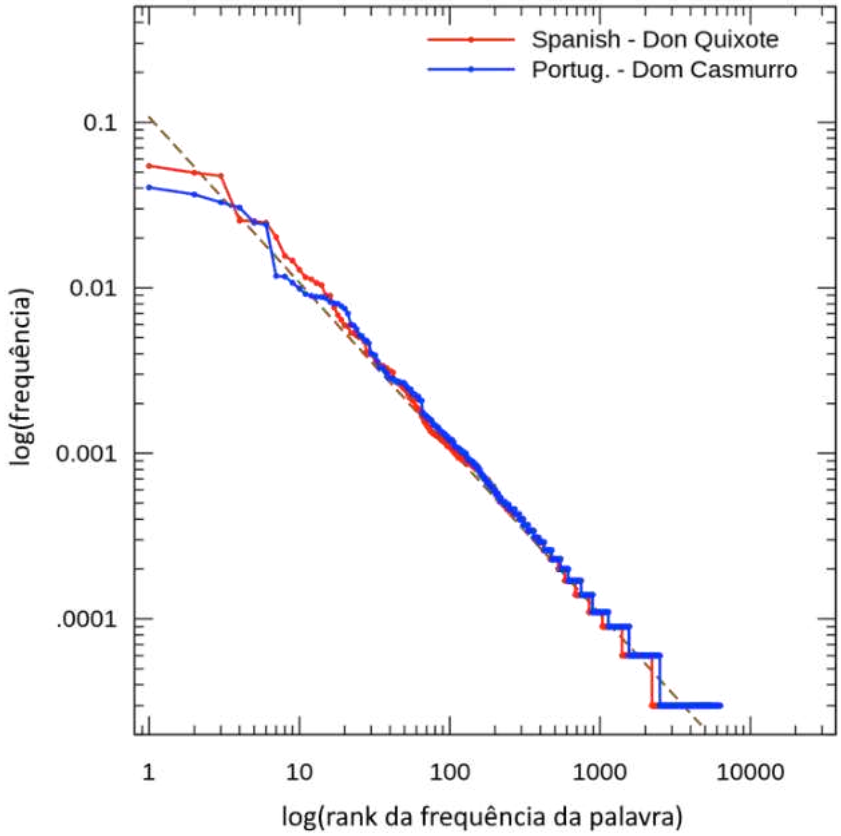
Tabela 2: POS tags implementadas pelo Spacy

POS tag	Significado
NOUN	substantivo
ADJ	adjetivo
PROPN	pronome
ADP	preposição
ADV	advérbio
VERB	verbo
PRON	pronome
SCONJ	conjunção subordinada
CCONJ	conjunção coordenativa
SPACE	espaço
INTJ	interjeição
PART	partícula
SYM	símbolo
DET	determinante
PUNCT	pontuação
NUM	número
AUX	auxiliar
X	outros

Fonte: https://spacy.io/models/pt (2023)

Em expressões linguísticas naturais, é comum observar um comportamento descrito pela Lei de Zipf. Segundo ela, a frequência de uma palavra em um corpus é inversamente proporcional à sua classificação na lista de frequências de palavras (ZIPF, 1936). Por exemplo, em um determinado texto, a palavra mais comum ocorrerá uma vez em cada 10 palavras, a segunda mais comum em cada 20 palavras, a terceira em cada 30 palavras e assim sucessivamente. A lei de Zipf pode ser observada em escritos de vários idiomas, como o português e o espanhol. Essa regularidade na ocorrência de palavras em um corpus pode ser usada para validar o resultado da coleta de palavras por um algoritmo de PLN, verificando se a frequência das palavras coletadas pelo algoritmo segue o padrão esperado pela lei de Zipf. Contudo, a lei de Zipf não pode ser usada como uma forma definitiva para determinar se um texto é composto por uma linguagem natural, pois é possível que um corpus possua pouca ou nenhuma correlação com a lei de Zipf e mesmo assim ser composto por uma linguagem natural (MORENO-SÁNCHEZ; CORRAL, 2016). A Figura 1 ilustra a frequência de cada palavra em função da sua posição no ranque de palavras mais frequentes de dois corpora. A linha pontilhada exibe os valores esperados pela Lei de Zipf.

Figura 1: Lei de Zipf aplicada a textos em português e em espanhol



Fonte: https://en.wikipedia.org (2024)

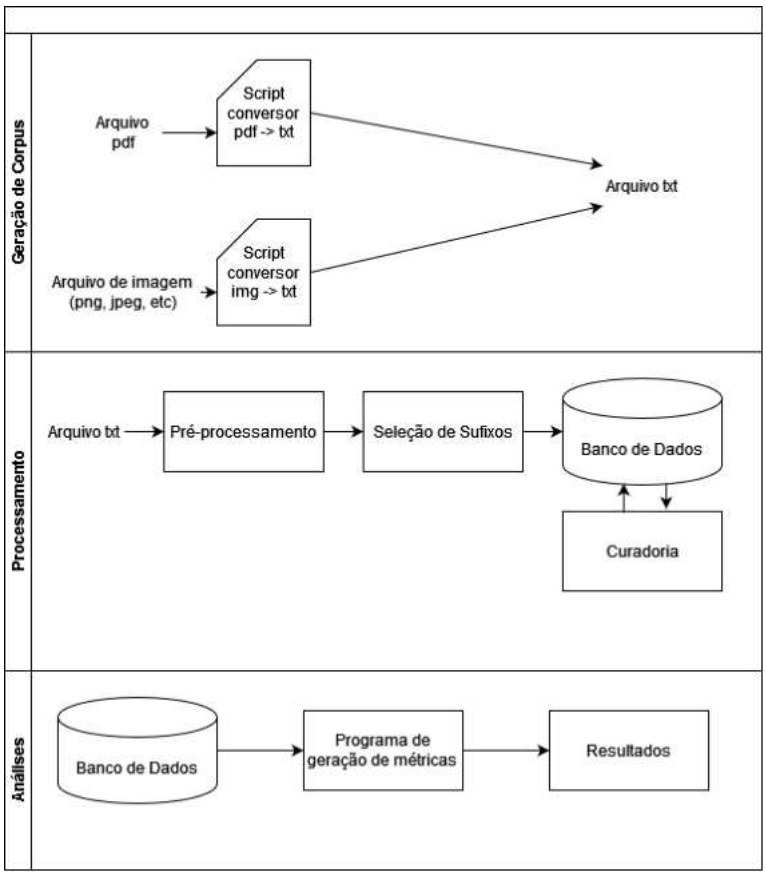
4 METODOLOGIA

Nesse capítulo será explicado a elaboração da ferramenta de coleta de dados morfológicos. Primeiro serão introduzidos os módulos que compõe a ferramenta e depois será detalhado o funcionamento de cada um desses módulos.

Dado um corpus na língua portuguesa, foi utilizada uma série de procedimentos para avaliar o seu conteúdo. Foi criado um processo, divido em três módulos, para criar, processar e, por fim, analisar corpora da língua portuguesa. No primeiro módulo, arquivos pdf ou de imagem (png, jpeg, etc) são lidos por um script conversor que retorna o conteúdo dos arquivos em um arquivo de texto. Este módulo não é necessário caso o corpus a ser processado já esteja no formato de texto simples. No segundo módulo, o de processamento, o programa lê o arquivo de texto retornado pelo módulo anterior como entrada. Depois, é feito o pré-processamento, que coleta informações sintáticas necessárias para a etapa de extração de sufixos, que é a etapa seguinte da execução. Os dados obtidos nas etapas anteriores são estruturados e salvos em um banco de dados, onde os dados obtidos podem ser avaliados quanto a sua corretude. Por fim, o módulo de análises faz consultas na base de dados montada e exibe as métricas relevantes.

O diagrama abaixo ilustra as etapas da metodologia utilizada. Os dois primeiros módulos serão mais bem descritos nas seções posteriores. O módulo de análises será mais bem explorado na seção 5.2.

Figura 2: Diagrama de etapas da metodologia utilizada



4.1 GERAÇÃO DE CORPUS

É comum livros, artigos científicos, periódicos etc. serem publicados em arquivos que não são arquivos de texto propriamente ditos. Formatos como PDFs e imagens podem ser usados para incluir ilustrações e tornar um texto mais visualmente interessante. Porém, o uso desses formatos de publicação sem nenhuma padronização pode atrapalhar a leitura dos textos por uma máquina.

A solução encontrada foi a criação de um script utilizando a biblioteca chamada Tesseract[4]. O Tesseract é uma ferramenta de OCR (Optical Character Recognition) de código aberto desenvolvida pelo Google que permite a extração de texto a partir de imagens ou documentos digitalizados. Esta é capaz de analisar imagens contendo texto e converter esse

[4] https://github.com/tesseract-ocr/tesseract

texto em formato legível por computador, um arquivo de texto simples no padrão Unicode[5]. Mais detalhes sobre essa ferramenta podem ser encontrados na publicação de (SMITH, 2013).

Antes que o texto gerado seja fornecido ao módulo de processamento, é importante que seja feito um processo de limpeza manual para remover qualquer ruído. É considerado como ruído qualquer dado na entrada do programa que não seja parte dos textos, por exemplo, elementos como números de páginas em livros, caracteres de controle não imprimíveis e código HTML extraído de artigos online podem afetar a qualidade do resultado. Vale ressaltar que frases mal construídas ou fragmentadas presentes no corpus também afetarão negativamente a leitura, pois a etapa de pré-processamento avalia a estrutura sintática da entrada, que é prejudicada por textos impróprios.

Podem ser utilizadas redações, cartas, livros ou simples listas de frases desconexas entre si. Porém, é importante que o conteúdo da entrada seja gramaticalmente coerente. Construções agramaticais como “Ele linguísticas para introduzirá disse onde”, apesar de estarem escritas em português, também irão afetar a precisão da fase de pré-processamento. Neste trabalho não foram implementados mecanismos de detecção e tratamento de construções agramaticais ou sem sentido semântico. O usuário precisa garantir que o texto fornecido como entrada não possui tais irregularidades.

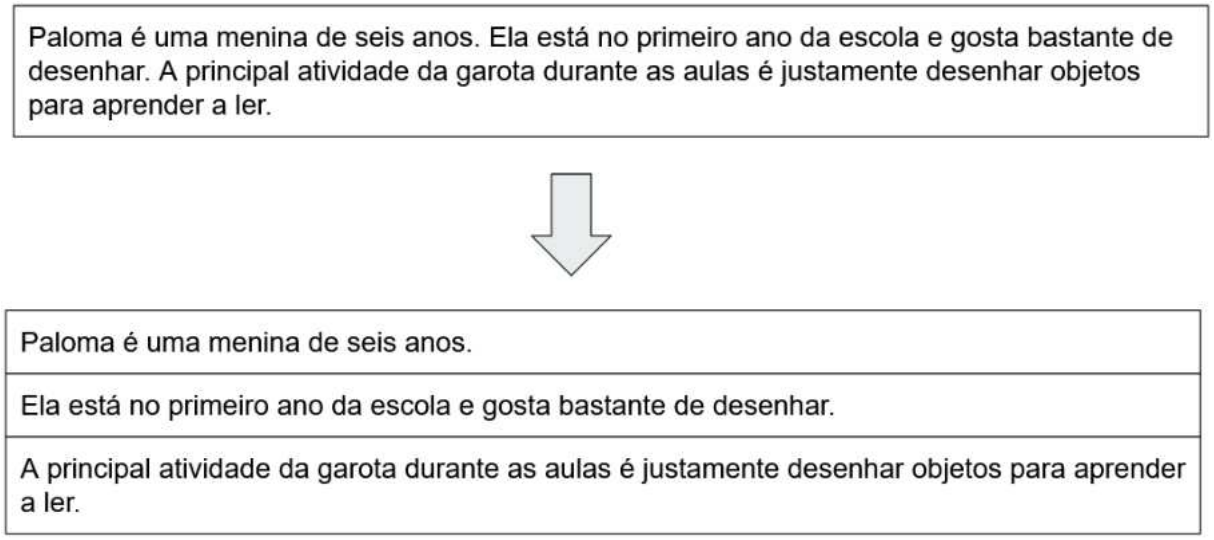
4.2 PROCESSAMENTO

O módulo de processamento deve receber o arquivo de texto em português gerado pelo módulo anterior como entrada. Após a leitura da entrada, o texto pode ser considerado apenas como uma sequência de caracteres, sem relação ou estrutura estabelecida. A etapa de pré-processamento define a estrutura do texto considerando as frases que compõem o texto, os elementos que compõem as frases e qual é o papel de cada elemento dentro de uma frase.

O pré-processamento é feito de forma incremental, onde a cada etapa são coletadas diferentes metadados sobre o texto lido. A primeira etapa do pré-processamento consiste em partir um texto nas frases que o compõem (Figura 3). Como frases bem construídas na língua portuguesa costumam ter um início e final bem padronizado, normalmente delimitados por sinais de pontuação, como interrogação, exclamação e ponto final, é razoavelmente simples realizar essa partição.

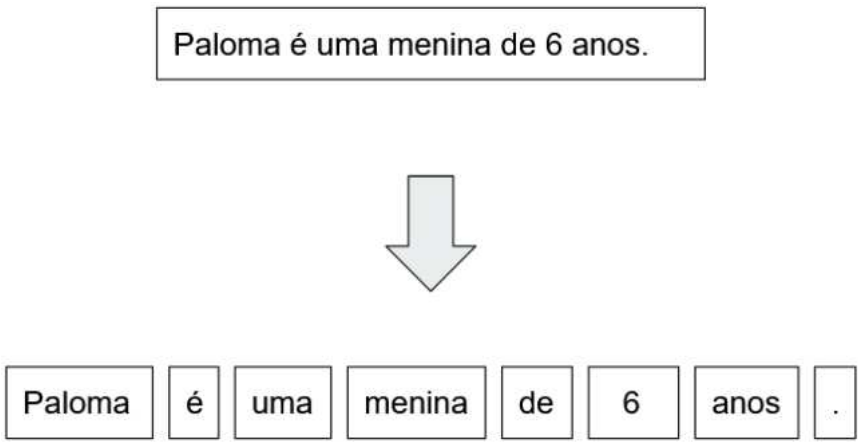
[5] https://www.unicode.org/standard/standard.html

Figura 3: Primeira etapa do pré-processamento



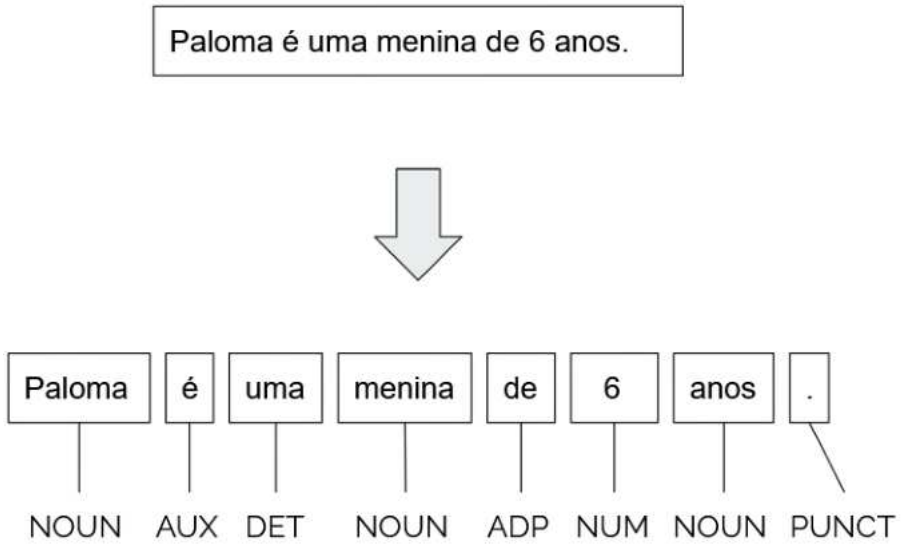
Após a primeira partição, é realizada a tokenização de cada frase, ilustrado na Figura 4.

Figura 4: Segunda etapa do pré-processamento



A realização das duas etapas anteriores permite uma visão mais granular do conteúdo do texto. Tendo listado cada componente da frase, é realizado o POS tagging para classificar cada um desses elementos, conforme exemplificado na Figura 5.

Figura 5: Terceira etapa do pré-processamento



Com o auxílio dos dados coletados no pré-processamento, é feita a coleta de sufixos das palavras. Inicialmente, foi adotado um algoritmo bem simples, utilizando uma lista de todos os sufixos da língua portuguesa para todas as classes gramaticais. Recebendo uma palavra como entrada, o algoritmo consiste em, basicamente, retornar os últimos n caracteres da palavra caso esses n caracteres formem um sufixo da lista, onde n é o comprimento do sufixo sendo comparado. Para a lista de sufixos utilizada nesse trabalho, o valor de n varia de 2 a 8.

Essa estratégia inicial apresentou dois problemas. O primeiro é o caso em que um sufixo está contido em outro. Por exemplo, o sufixo -ão está contido dentro de -eirão (ambos do tipo aumentativo). Dessa forma, não fica claro qual sufixo deve ser atribuído à uma palavra que termina em "eirão". Uma solução possível para essa ambiguidade seria considerar sempre o maior sufixo encontrado. Isso resolveria a maioria dos casos, porém, essa solução é imprecisa. Alguns termos, como a palavra "feirão" parecem ter o sufixo -eirão, mas essa palavra possui raiz feir- e sufixo -ão.

Outro problema com esse algoritmo, é que ele não considera que o sufixo de uma palavra pode variar dependendo da sua classe gramatical. Por exemplo, a palavra "casarão", se ela for o aumentativo do substantivo "casa", o sufixo da palavra é -arão, mas se essa mesma for uma conjugação do verbo "casar", o sufixo que deve ser considerado é -ão.

Para resolver o primeiro problema encontrado, ao invés de comparar todos os sufixos possíveis com o final da palavra, foi utilizado o stemmer para extrair a raiz da palavra. Depois, a raiz extraída é comparada com a palavra completa. Todo texto da palavra que fica após a raiz foi considerado como sendo o sufixo.

Como a implementação do stemmer pode não ser perfeita, é feito uma verificação se o sufixo extraído existe na lista de sufixos da língua portuguesa. Em caso positivo, o sufixo encontrado é retornado, senão, o sufixo é dado como errôneo e a palavra é considerada como não tendo um sufixo. Essa solução resolve a ambiguidade que ocorre em sufixos contidos em outros sufixos, mas ele está sujeito às imprecisões do stemmer, que pode extrair uma raiz maior ou menor que o valor correto, o que é mais comum com vocábulos e inflexões pouco utilizadas, irregulares ou ambíguos. Esse comportamento pode ser encontrado com a palavra "vozeirão" cuja raiz o NLTK identifica como sendo voze-, o que implicaria em um sufixo -irão sendo que o sufixo correto é -eirão. Outro exemplo dessa imprecisão é a palavra "Japão", na qual o stemmer do NLTK identifica erroneamente as duas últimas letras do nome do país como sendo o sufixo aumentativo -ão.

Para tratar o problema de palavras que possuem sufixos diferentes dependendo da sua classe gramatical, foi feita uma modificação na lista de sufixos. A lista foi substituída por uma lista de três listas, onde cada lista desta estrutura contém os sufixos que podem ocorrer nos substantivos, verbos e advérbios. Também foi necessário modificar a entrada do algoritmo. Além de receber a palavra cujo sufixo deve ser extraído como entrada, ele também recebe a sua classe gramatical. Nesta implementação, é considerada a classe gramatical das palavras estimadas pelo Spacy.

Para obter um resultado mais acurado, o algoritmo extrai a raiz da palavra com o stemmer do NLTK e compara o segmento da palavra após a raiz encontrada com a lista de sufixos que podem acontecer na classe gramatical em questão. Na lista de sufixos que foi montada, cada sufixo só está associado a uma única classe gramatical. Um sufixo nominal não pode ocorrer em um verbo, um sufixo verbal não pode ocorrer em um advérbio, e assim sucessivamente. Dessa forma, uma palavra fica restrita apenas aos sufixos de sua classe gramatical, porém, além da imprecisão do *stemmer*, agora é preciso considerar também a imprecisão da classificação gramatical do Spacy. O algoritmo final segue as seguintes etapas:

1. Mudar todos os caracteres maiúsculos para minúsculos.
2. Extrair radical da palavra usando o Stemmer.
3. Definir toda a fração da palavra depois do radical como um possível sufixo.
4. Se o possível sufixo estiver contido na lista de sufixos permitidos da respectiva classe gramatical da palavra da qual ele foi extraído, retornar o possível sufixo.
5. Caso contrário, retornar que a palavra não possui sufixo.

4.3 IMPLEMENTAÇÃO

A princípio, o Spacy seria uma ferramenta razoavelmente simples de ser utilizada, porém houve complicações em relação ao uso de memória. A biblioteca é capaz de consumir textos pequenos, como uma carta, um artigo ou um livro com considerável agilidade e facilidade. Porém, ao tentar processar um corpus inteiro com mais de 13 milhões de caracteres, o programa falhou em sua execução. Monitorando o consumo de recursos do sistema, ficou evidente que, durante a etapa de processamento do Spacy, o consumo de memória aumentou linearmente até atingir todos os 16 GiB de memória RAM disponíveis na máquina e, subsequentemente o programa falhou por falta de memória. Outros testes feitos na mesma máquina mostraram que o programa é capaz de ler cerca de cento e vinte mil caracteres antes de consumir toda a memória disponível.

Durante a implementação, os testes iniciais foram feitos com um corpus de notícias, além dos corpora de livros infantis e de cordéis, que serão descritos no decorrer do trabalho. Idealmente, o programa não deveria ser fortemente limitado pela quantidade de memória. Não foi possível ler sequer um décimo do corpus de notícias tendo à disposição 16 GiB de memória, uma quantidade relativamente comum em computadores pessoais. Para resolver esse problema, foi criado um algoritmo que particiona o texto de entrada em partições com comprimento máximo definido e que processa cada uma dessas partições individualmente.

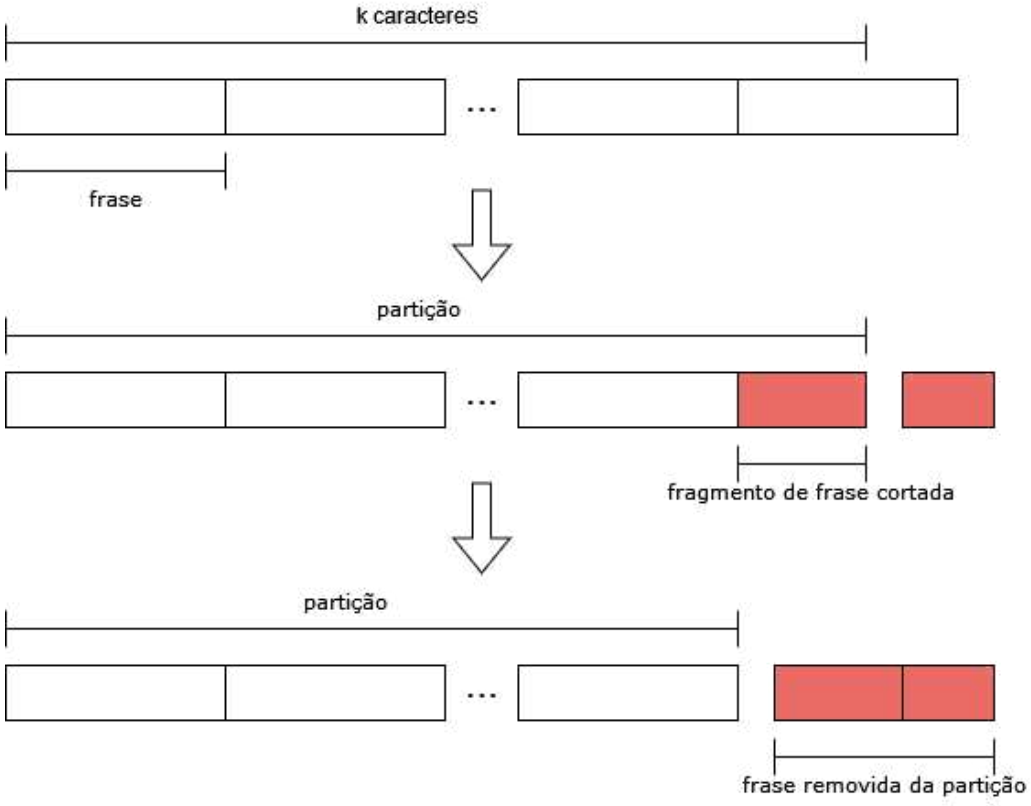
No algoritmo de particionamento, o corpus é passado como um parâmetro no formato de texto. O número máximo de caracteres de cada partição, que será considerado como o "tamanho" da partição, deve ser especificado pelo usuário, cujo valor deve ser definido considerando a quantidade de memória disponível na máquina. Não há uma regra especificando o melhor tamanho de partição para uma determinada máquina, sendo necessária a experimentação e avaliação pelo usuário para determinar o valor mais adequado para cada cenário. Ao longo deste trabalho, o tamanho máximo de cada partição foi definido em cem mil caracteres. Tal valor foi suficiente para comportar o programa em 16 GiB de memória sem causar uma sobrecarga significativa na etapa de particionamento.

O particionamento é feito por iterações, onde cada iteração processa uma partição do corpus. As partições são sequenciais, cada uma começa onde a anterior termina. Como o início e final de frases nem sempre alinham com o início e final de cada partição, não é possível definir o comprimento das partições como uma constante. É preciso garantir que palavras e frases não sejam fragmentadas no processo, pois cada partição é processada independentemente das

demais. Sendo assim, para processar uma partição, o algoritmo processa todos os k caracteres após o final da última partição, onde k é o tamanho máximo de cada partição. Para o final da partição não cortar o meio de uma frase, o final de cada partição é movido para o final da frase anterior, deixando a frase que foi fragmentada para ser processada inteiramente pela partição seguinte. Esse processo se repete até o corpus inteiro ser lido. O algoritmo de particionamento segue as seguintes etapas:

1. Definir início da partição no primeiro caractere após a partição anterior (primeiro caractere do corpus caso seja a primeira parição).
2. Definir o final da partição no k-ésimo caractere depois do início da partição.
3. Pré-processar a partição.
4. Caso a partição não contenha o final do corpus, descartar a última frase pré-processada.
5. Redefinir o final da partição para o último caractere da última frase não descartada.
6. Caso a partição não contenha o final do corpus, voltar para a etapa 1.

Figura 6: Algoritmo de particionamento



Após a implementação do algoritmo de particionamento, o programa foi capaz de processar os três corpora usados nesse trabalho adequando o consumo de memória à capacidade da máquina. Porém, conforme se tornou possível processar quantidades maiores de texto, outros problemas se tornaram aparentes:

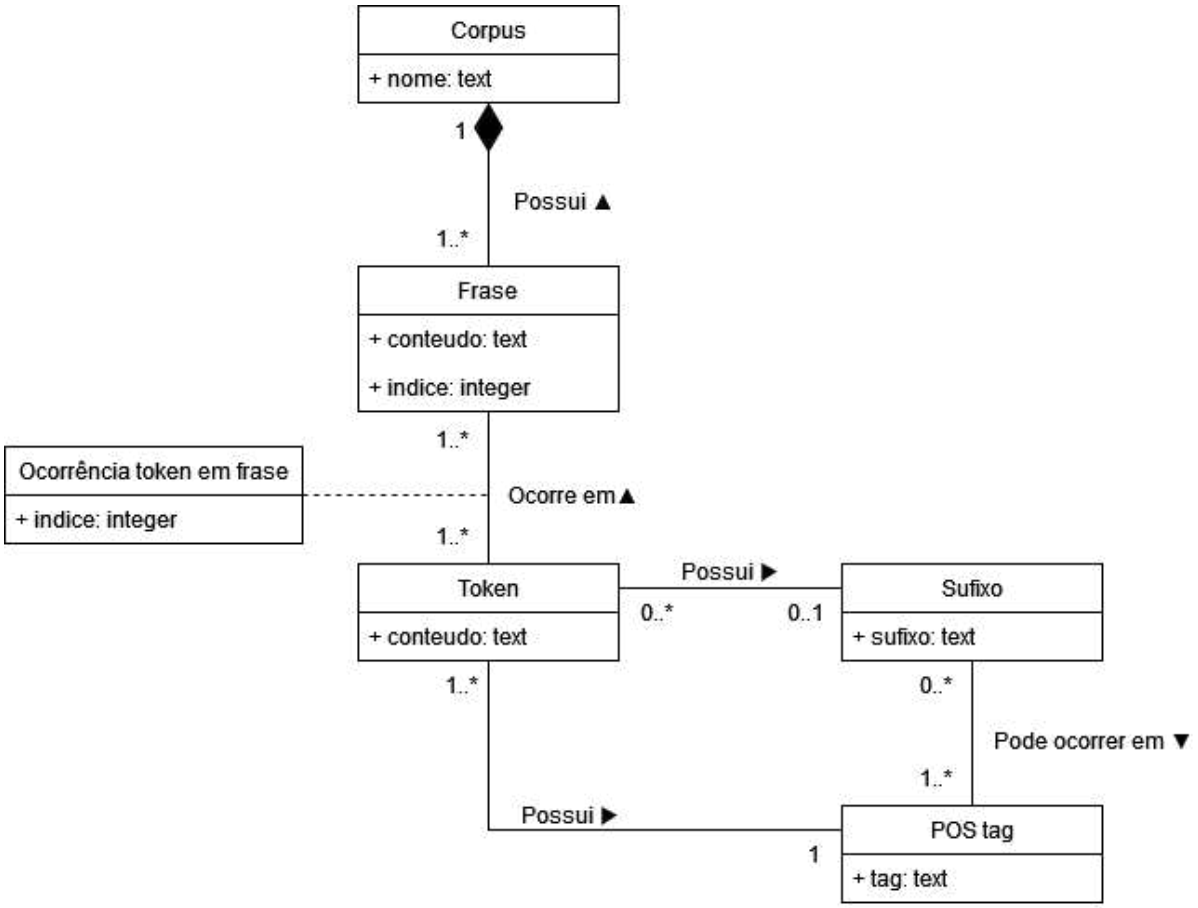
1. O tempo de processamento se tornou uma questão a se considerar. Com corpora de tamanho reduzido, é possível processar os textos e depois gerar as métricas em poucos minutos. Porém, conforme a quantidade de textos aumenta, se torna impraticável ter que reprocessar um corpus toda vez que for necessário coletar uma nova métrica, sendo que todas podem ser coletadas em uma única execução.
2. A precisão dos dados coletados pode ser maior ou menor dependendo do corpus processado, mas, de qualquer forma, seria interessante o usuário ser capaz de inspecionar e corrigir erros de coletas de métricas atualizando o banco de dados diretamente.
3. Não havia sido previstas formas de comparar corpora diferentes pois, em cada execução, apenas um corpus é processado e tem as suas métricas coletadas individualmente.

Pensando nesses três problemas citados, um banco de dados foi adicionado no módulo de processamento. Após processar um corpus no formato de texto, o módulo armazena o resultado do processamento em um banco SQLite. O SQLite foi o Sitema de Gerenciamento de Banco de Dados Relacional (SGBDR) escolhido para este trabalho devido à sua simplicidade e portabilidade. Ao contrário de outros SGBDRs, o SQLite não exige um servidor para operar, os dados ficam salvos em um único arquivo que pode ser movido, duplicado e alterado com facilidade.

A Figura 7 contém um diagrama de classes, modelado seguindo as especificações UML[6], representando o banco de dados utilizado. Para fins de documentação, foi incluído no Apêndice B um diagrama entidade relacionamento (ER) que representa o banco em um nível lógico.

[6] https://www.omg.org/spec/UML/2.5.1/About-UML

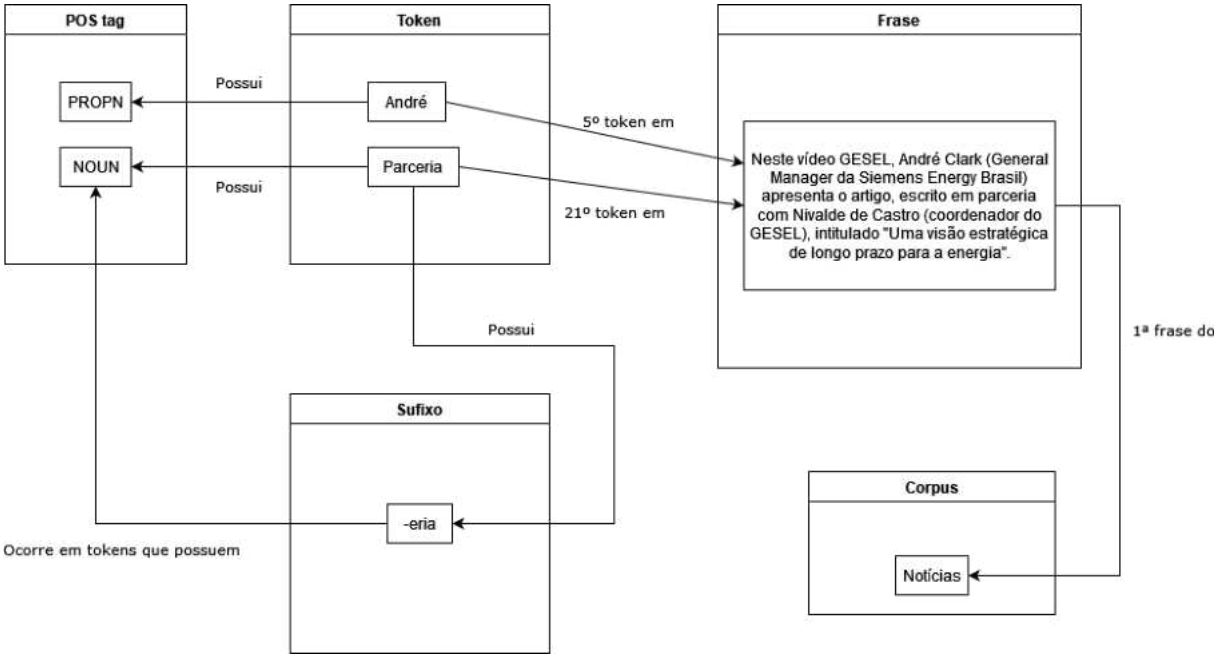
Figura 7: Diagrama de classes do banco de dados



O banco de dados foi estruturado para armazenar dados de diversos corpora. Um corpus é identificado pelo seu nome, que é definido arbitrariamente pelo usuário. Um corpus é composto por diversas frases, que são identificadas pelo corpus ao qual elas pertencem e o índice delas dentro do corpus. Por exemplo, a primeira frase do corpus X, 13ª frase do corpus Y (sendo X e Y dois nomes de corpora distintos). Diferentemente das frases, múltiplas ocorrências de token idênticos correspondem à um único registro na tabela de tokens. Tokens são identificados pelo seu conteúdo e pelo seu POS tag. Por exemplo, o substantivo “casarão” é considerado um token distinto do verbo “casarão”. Um token pode ou não possuir um sufixo, sendo que um sufixo só pode ocorrer em uma palavra cujo POS tag está na sua lista de POS tags permitidos. Os sufixos que estão presentes na tabela de sufixos serão chamados de “sufixos validados”, pois ela contém apenas sufixos cuja veracidade foi conferida manualmente. Visando explorar possíveis sufixos que não foram contemplados na lista de sufixos válidos, o programa insere qualquer possível sufixo encontrado na tabela de tokens, seja ele validado ou não. Para diferenciar ocorrências de sufixos validados e não validados, basta comparar a tabela

de tokens com a tabela de sufixos. A Figura 8 ilustra um exemplo das entidades presentes no banco de dados.

Figura 8: Exemplo do modelo do banco de dados



Salvar os dados coletados dos corpora em um banco estruturado possui três vantagens. Primeiramente, é dispensada a necessidade de reprocessar o corpus toda vez que for necessário coletar novas métricas. Após o processamento inicial, todos os dados coletados ficam salvos no banco para referências futuras. A segunda vantagem é a organização da informação, utilizar um banco de dados modelado e documentado facilita o acesso e modificações futuras nos dados. Por fim, essa forma de armazenamento dos dados viabiliza correções feitas pelo usuário posteriores ao processamento.

Como é esperado que o processamento de um corpus não seja perfeitamente preciso, o usuário pode aplicar correções ou modificações inserindo ou atualizando informações diretamente no banco. Caso, um usuário queira, por exemplo, incluir novos sufixos para serem procurados nas palavras do corpus, ele pode fazer inserções desses sufixos na respectiva tabela do banco de dados antes de realizar o processamento do corpus. Este procedimento, no entanto, exigirá o reprocessamento de todos os corpora já presentes no banco. Se o usuário identificar um erro no resultado do processamento, como uma palavra com um sufixo ou POS tag errados, ele pode realizar as devidas correções posteriormente, sobrescrevendo os dados inválidos de forma manual.

Ocorrências de sufixos em tokens podem ser corrigidas antes do processamento, pois o programa não altera as ocorrências que já existem no banco. Como diversas ocorrências de uma única palavra referenciam uma única entrada na tabela de palavras, correções aplicadas à uma palavra são automaticamente aplicadas à todas as ocorrências desta palavra em todos os corpora, o que facilita a manutenção dos dados, além de permitir o usuário consultar em quais corpora um token ocorre. O programa nunca sobrescreve os dados que já estão no banco, logo, quando ele processa um novo corpus, as correções feitas anteriormente pelo usuário prevalecem.

A modelagem do banco permite a correção de pseudosufixos. Antes de processar os corpora, o usuário pode popular a tabela de tokens com palavras que sabidamente não possuem sufixo. Ao processar os corpora posteriormente, o programa irá considerar a lista pré-preenchida. Os dados de sufixos e de tokens são únicos para todos os corpora no banco, logo, qualquer alteração nesses dados será aplicada a todos os corpora no banco. O usuário também pode corrigir pseudosufixos após o processamento do corpus, ajustando a tabela de tokens criada pelo módulo de processamento.

5 APLICAÇÃO DA FERRAMENTA

Neste capítulo serão apresentados os três corpora que foram processados pela ferramenta. Os corpora possuem assuntos variados, um de notícias do setor elétrico, um de livros infantis e um de literatura de cordéis. Posteriormente, serão apresentados os resultados obtidos pelo processamento destes corpora, onde os três serão comparados entre si.

5.1 CORPORA

Inicialmente, criamos o banco de dados usando um corpus formado por resumos de notícias do setor elétrico (OLIVEIRA, 2023). As notícias presentes nesse corpus foram extraídas da página web Informativo Eletrônico do Setor Elétrico (IFE), que é um portal de notícias do Grupo de Estudos do Setor Elétrico (GESEL) da Universidade Federal do Rio de Janeiro (UFRJ).

Textos jornalísticos e acadêmicos tendem a aderir à forma padrão da língua portuguesa e serem mais formais do que outras fontes. Como foi explicado na seção 4.1, desvios de linguagem podem interferir no processamento do corpus. Por isso, um corpus de resumo de notícias com textos curados e revisados se mostrou uma opção mais adequada para fazer a primeira coleta de dados do que uma fonte de textos menos padronizada, como seria o caso de corpora de mensagens de texto ou de publicação em redes sociais. Neste estudo não foi considerado o corpus inteiro, mas apenas uma fração dos textos disponíveis contendo 87550 frases.

Neste trabalho, foram criados mais dois corpora para serem processados pela ferramenta. O primeiro foi criado com o conteúdo de nove livros infantis. Processar textos de livros infantis tem um alto potencial para agregar valor ao ensino infantil, pois pode resultar em conhecimentos que permitem auxiliar o entendimento de como as crianças desenvolvem sua escrita e sua fala. Esta é a principal ideia para querermos usar um corpus de livros infantis como um teste de validação para nossa ferramenta.

O corpus infantil foi construído a partir de textos extraídos de livros infantis no Portal Domínio Público[7]. Primeiramente, foram baixados apenas livros no formato PDF nos quais seria possível copiar o seu texto diretamente do arquivo. Em seguida, foi desenvolvido um script

[7] http://www.dominiopublico.gov.br

em Python que consegue ler arquivos deste tipo e cria um arquivo de texto no formato de entrada aceito pela ferramenta, como já foi apresentado na seção 4.1.

Antes de inserirmos o arquivo texto criado na ferramenta para alimentar o banco e gerar as análises, foi executada uma etapa de validação humana, que acabou ressaltando erros de captura de caracteres, o que nos levou a recorrer a ajustes manuais para remover estes erros ortográficos. Como a quantidade de caracteres total do arquivo se aproximou de 900 mil, optamos por não fazer esta correção de forma manual, mas sim utilizando a ferramenta de edição de textos Google Docs[8] para identificar e corrigir estes erros. Apesar desta ferramenta ser muito boa, em alguns casos ela sugeria uma alteração incorreta para o texto original da palavra ou frase. Mesmo assim, mantivemos estas correções por não ser tão fácil de corrigirmos estes casos que, apesar de poucos em relação às sugestões corretas, ainda eram muitas para um humano corrigir manualmente. É apropriado mencionar que, apesar de não ter sido utilizada durante a elaboração do script, o Google Docs API[9] pode ser utilizado diretamente no código. Após estas etapas, o arquivo texto foi inserido para ser processado pela ferramenta e foram gerados os resultados.

O último corpus foi criado com textos de literatura de cordéis. Antes de descrever o processo de extração é importante fazer uma breve contextualização sobre o que são cordéis. Muito populares na Europa, os cordéis começaram a se popularizar no Brasil durante o período colonial, quando começaram a ser trazidos pelos portugueses (MENESES, 2019). Os cordéis são escritos em forma de poemas e utilizando xilogravuras e receberam este nome porque eram colocados para serem vendidos pendurados em cordas ou cordéis.

Após utilizar o mesmo site dos livros infantis para obtermos os cordéis, que também foram utilizados nove livros, partimos para o processo de tratamento. Esse processo, inicialmente, consistiria em utilizar o mesmo script de extração utilizado para os livros infantis. Porém, como a maior parte dos cordéis disponibilizados no site de domínio público não possuem suporte para o formato digital[10], tivemos que extrair seu texto.

Como os arquivos dos cordéis são no formato PDF e a biblioteca precisa que a entrada sejam imagens, tivemos que utilizar a etapa de geração de corpus para transformar cada página

[8] https://www.google.com/intl/pt-BR/docs/about/

[9] https://developers.google.com/docs/api/reference/rest?hl=pt-br

[10] Neste caso, entende-se formato digital como um arquivo PDF no qual se consegue selecionar o texto via mouse ou software, ou seja, há alguma codificação para os caracteres.

dos PDFs em imagens e posteriormente em textos. Inicialmente isso foi feito utilizado o site iLovePDF[11], que possui uma funcionalidade para executar esta tarefa. Porém, depois foi descoberto que o OCR também é capaz de analisar imagens em pdf, o que dispensaria o uso do site.

Após esse processo e do desenvolvimento do script, pudemos gerar nosso arquivo texto com os cordéis. No corpus dos livros infantis enfrentamos a dificuldade relacionada aos erros de identificação de caracteres, e o mesmo ocorreu aqui com uma frequência maior, entretanto foi utilizado novamente o Google Docs para remover estes erros.

É importante observar que, devido à evolução da língua ao longo do tempo, as regras gramaticais e ortográficas passaram por diversas mudanças desde a época em que os textos foram originalmente escritos, especialmente durante a época colonial. Além disso, os modelos de aprendizado de máquina, como os utilizados em ferramentas de OCR, são treinados com base em dados disponíveis no meio digital, que refletem predominantemente o uso contemporâneo do idioma e que isso pode afetar o desempenho tanto da ferramenta OCR quanto do Google Docs.

Antes de começarmos a seção 5.2, iremos introduzir algumas informações obtidas do processamento de cada corpus. Fazendo uma breve avaliação, foi possível notar alguns erros na coleta de sufixos. Por exemplo, o sufixo aumentativo -ão foi erroneamente atribuído aos substantivos “visão”, “difusão”, “função” e “expansão”, o sufixo coletivo -ada foi atribuído à palavra “década” e o sufixo denotativo de qualidade -or foi atribuído ao termo “setor”. Não foram encontradas falhas significativas na coleta dos tokens. Alguns outros erros foram encontrados na forma como as frases foram separadas no corpus, mas tais erros foram provocados por impurezas localizadas no próprio corpus, como aspas que foram abertas e nunca fechadas, e metadados do texto que foram inseridos no próprio texto.

Erros assim foram encontrados em todos os três corpora e, para corrigir o máximo de erros possível, foi preciso realizar uma curadoria manual no resultado do processamento e aplicar as devidas correções. Vale notar que, devido à forma que o banco de dados que armazena o resultado foi modelado, as correções realizadas nos sufixos coletados neste corpus serão aplicadas automaticamente aos corpora que forem processados no futuro, o que torna o processo de revisão manual gradualmente mais simples.

[11] https://www.ilovepdf.com/pt/pdf_para_jpg

5.2 RESULTADOS E VALIDAÇÃO

Daremos início à etapa de validação do processo e do banco de dados. Tendo processado os três corpora pela ferramenta, foi possível realizar análises com os dados coletados. Esses dados serão utilizados para responder as perguntas na Tabela 3, que foram criadas para validar o trabalho.

Tabela 3: Perguntas utilizadas para gerar resultados para a validação.

NÚMERO	PERGUNTA
1	Frequência em que palavras das classes gramaticais NOUN, VERB e ADV ocorrem em cada corpus.
2	Frequência total em que cada palavra ocorre em cada corpus.
3	Frequência em que os sufixos aparecem em cada corpus.
4	Frequência em que os radicais aparecem em cada corpus.
5	Quantidade de radicais por classe gramatical em cada corpus.
6	Razão entre o total de palavras com sufixo e o total de palavras por corpus.
7	Dentre as palavras que podem possuir sufixos (verbos, advérbios e substantivos), quantas de fato possuem um sufixo.
8	Quantidade de radicais que possuem sufixos diferentes em cada corpus.

Para obtermos as respostas das perguntas apresentadas na Tabela 3 foram executadas duas consultas ao banco de dados via SQL: a primeira gerou a tabela de dados, que iremos chamar de *Sufixos Válidos*, e que contém a palavra, seu sufixo, sua classe gramatical e o seu corpus. Além disso, esta mesma tabela só possui palavras cujos sufixos são validados, ou seja, sufixos que foram listados manualmente por um humano. Já a segunda tabela, que denominamos de *Geral*, contém os mesmos atributos da tabela *Sufixos Válidos,* porém contém todas as palavras, independentemente se possuem um sufixo válido ou não. O conteúdo da tabela *Sufixos Válidos* e da tabela *Geral* estão descritos na Tabela 4 e na Tabela 5 respectivamente.

Tabela 4: Contagens na tabela filtrada.

TABELA	CORPUS	QUANTIDADE DE FRASES	QUANTIDADE DE TOKENS	QUANTIDADE DE TOKENS DISTINTOS
Sufixos Válidos	Notícias	67.376	163.159	3.730
Sufixos Válidos	Livros infantis	4.731	8.962	1.835
Sufixos Válidos	Cordéis	52	97	65

Tabela 5: Contagens na tabela geral.

TABELA	CORPUS	QUANTIDADE DE FRASES	QUANTIDADE DE TOKENS	QUANTIDADE DE TOKENS DISTINTOS
Geral	Notícias	87.550	2.540.805	60.269
Geral	Livros infantis	8.629	202.858	14.441
Geral	Cordéis	145	2.551	822

5.2.1 Frequência de palavras

Os gráficos a seguir mostram os resultados gerados para o corpus de notícias para as perguntas:

- Pergunta 1 - Frequência em que palavras das classes gramaticais substantivo (NOUN), verbo (VERB) e advérbio (ADV) ocorrem em cada corpus.
- Pergunta 2 - Frequência total em cada palavra ocorre em cada corpus.

Ambos os gráficos mostram a frequência em que as palavras ocorrem, o primeiro em relação à tabela Sufixos Válidos e o segundo à tabela Geral. Convém destacar que optamos por mostrar apenas 2% das palavras na Figura 9 e 0,2% das palavras na Figura 10, para facilitar a legibilidade.

Figura 9: Frequência de ocorrência das palavras com sufixo - Corpus Notícias.

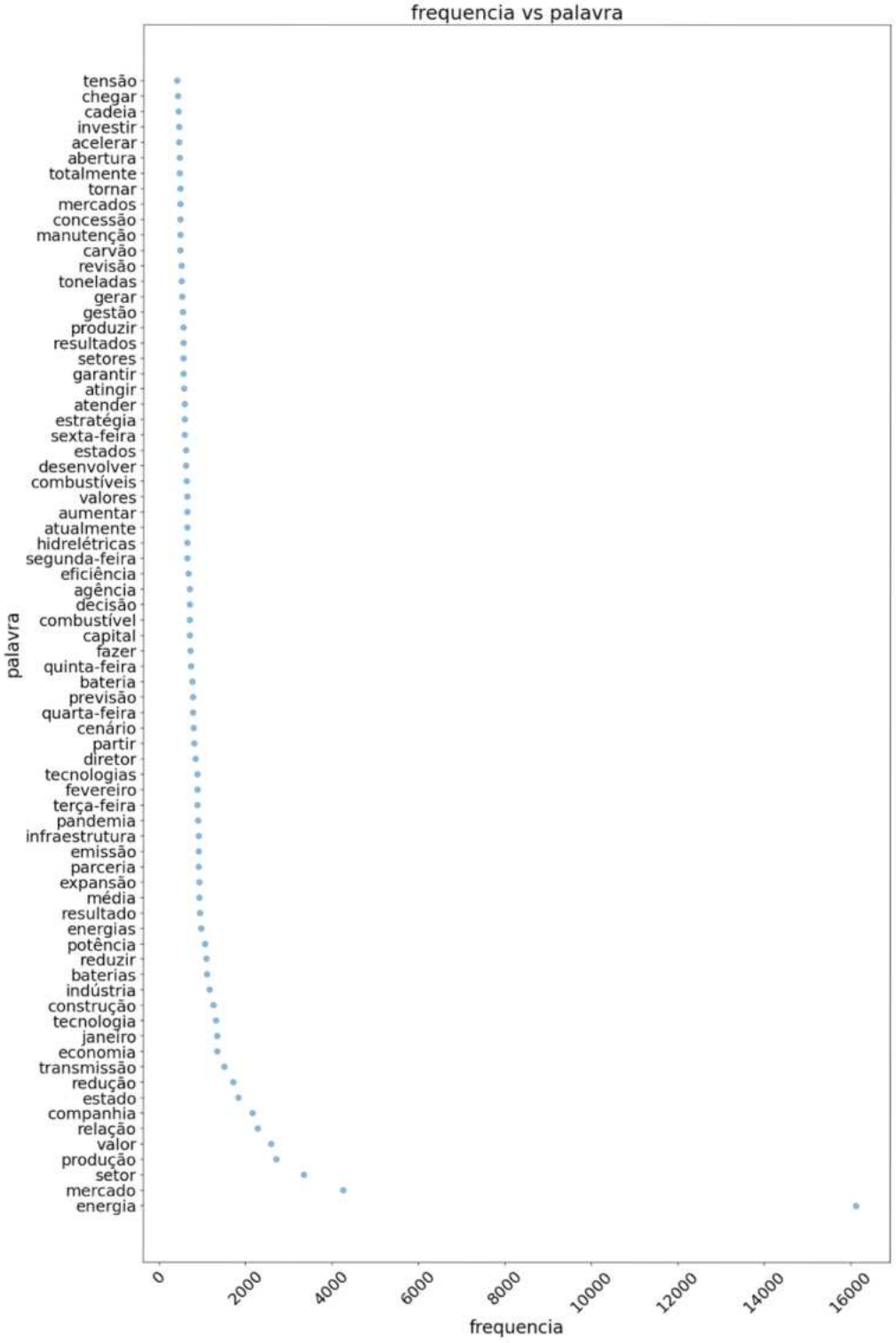
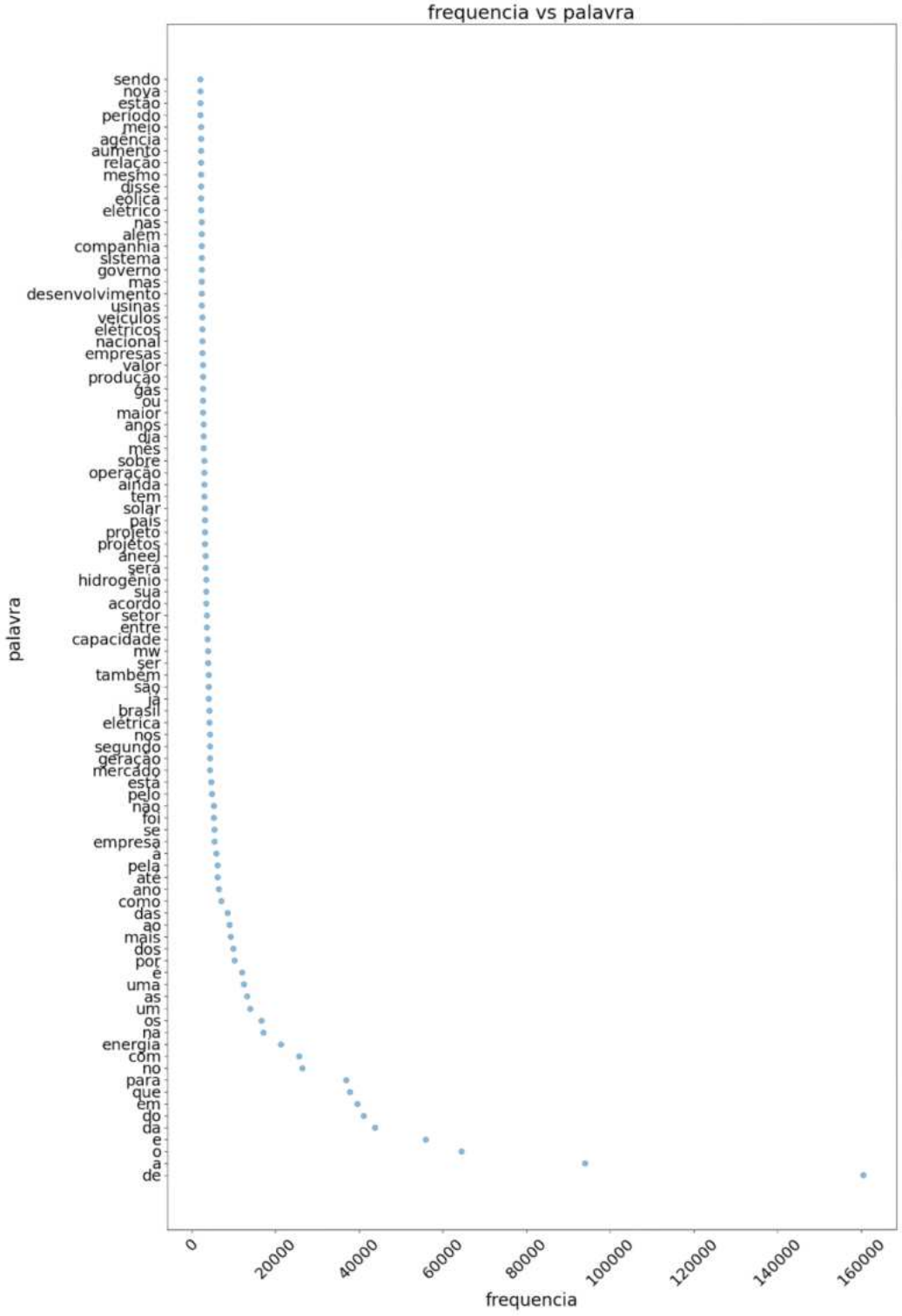


Figura 10: Frequência de ocorrência de todas as palavras - Corpus Notícias.



Analisando estas figuras, podemos perceber que independente do escopo de dados que estamos trabalhando houve uma tendência a uma curva exponencial. Uma outra percepção é que na Figura 10 os artigos e preposições assumem as primeiras posições no ranqueamento da

frequência, o que é algo de se esperar e que também é um forte indicador de que a ferramenta está com uma boa precisão em relação ao seu objetivo. É possível observar no gráfico a presença de várias *stopwords*. *Stopwords* são palavras frequentemente usadas que são filtradas em análises de texto por terem pouco significado analítico, como preposições e artigos. A decisão de manter as *stopwords*, originou-se da ideia de que seria interessante visualizar que estas sempre ocorrem em uma frequência superior à das outras palavras.

Para o corpus de livros infantis, podemos ver que a Figura 11 e a Figura 12 seguem o mesmo padrão do corpus de notícias, e que é um indicativo de que a base está bem tratada e que o programa mantém o padrão de resposta. Vale observar a diferença das palavras que mais ocorrem quando comparamos os dois corpora e que mesmo sendo distintas ainda mantém um movimento exponencial. As Figura 11 mostra 4% do total de palavras, enquanto a Figura 12 mostra 0,6% do mesmo total.

Figura 11: Frequência de ocorrência das palavras com sufixo - Corpus Livros Infantis.

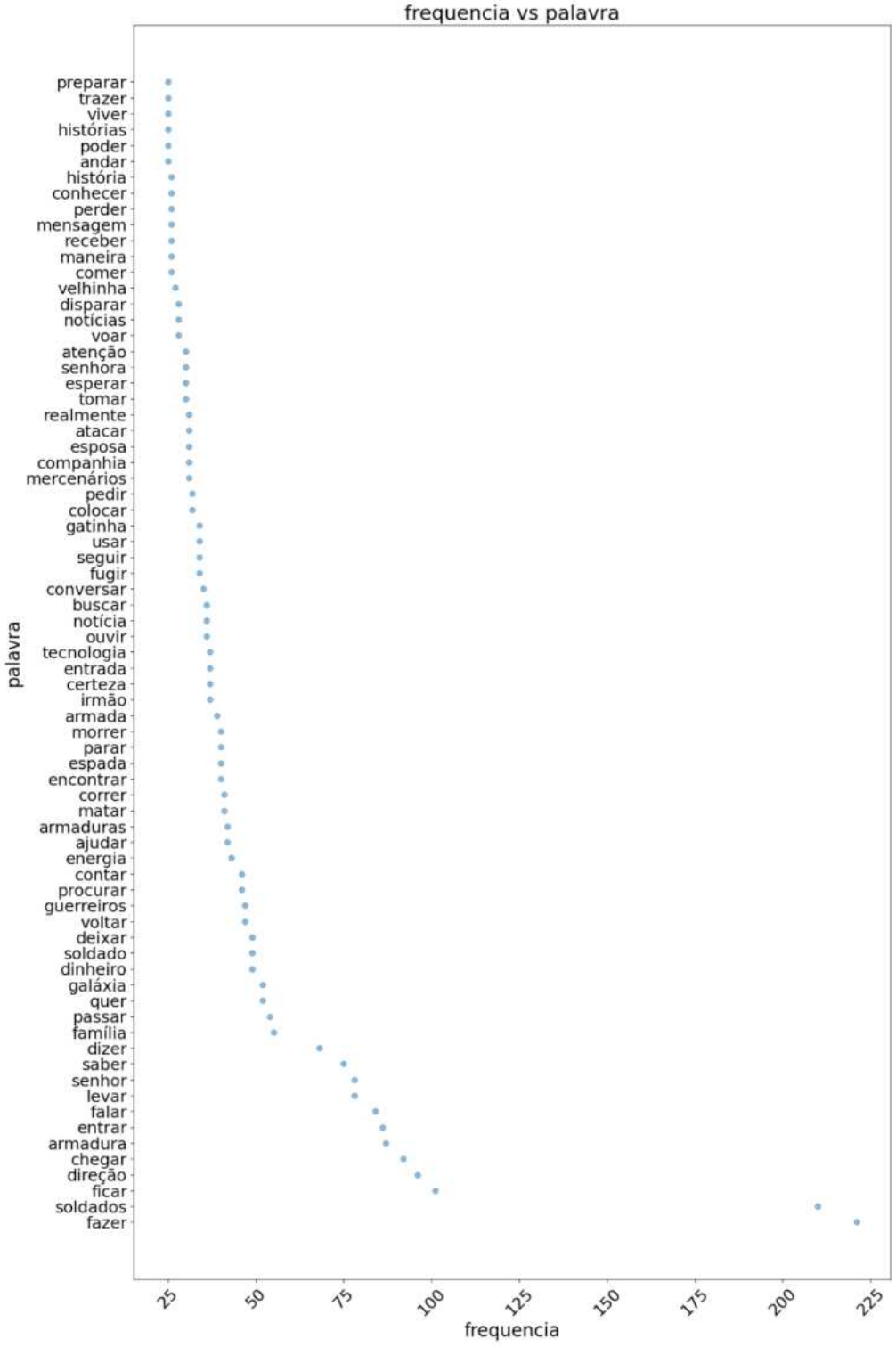
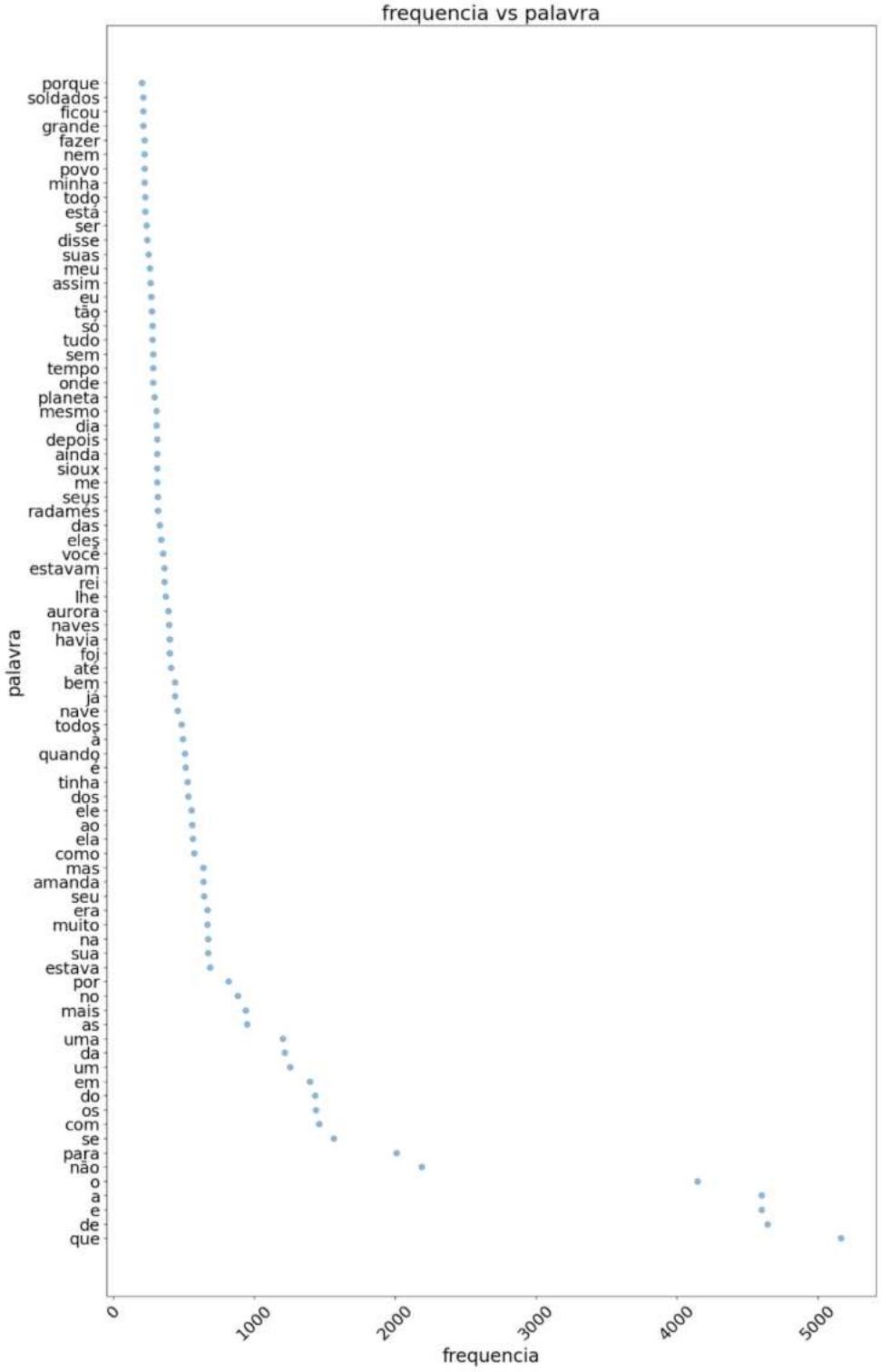


Figura 12: Frequência de ocorrência de todas as palavras - Livros Infantis.



No corpus dos cordéis, Figura 13 e Figura 14, é possível perceber a mesma tendência que os demais, mesmo havendo uma grande diferença na quantidade de vezes em que uma palavra ocorreu. A Figura 13 mostra 100% do total de palavras, enquanto a Figura 14 mostra 9% do mesmo total.

Observando os resultados mostrados na Figura 14, há a ocorrência de alguns caracteres especiais que são classificados como palavras, mas que na verdade não são. Este erro pode ocorrer devido ao funcionamento da lógica de classificação das palavras pela biblioteca Spacy, que quando somado às falhas de reconhecimento de caracteres pela ferramenta do Tesseract OCR acaba gerando este resultado errôneo. Devido a esta possibilidade, estas ocorrências devem ser desconsideradas para análises.

Figura 13: Frequência de ocorrência das palavras com sufixo - Corpus Cordéis.

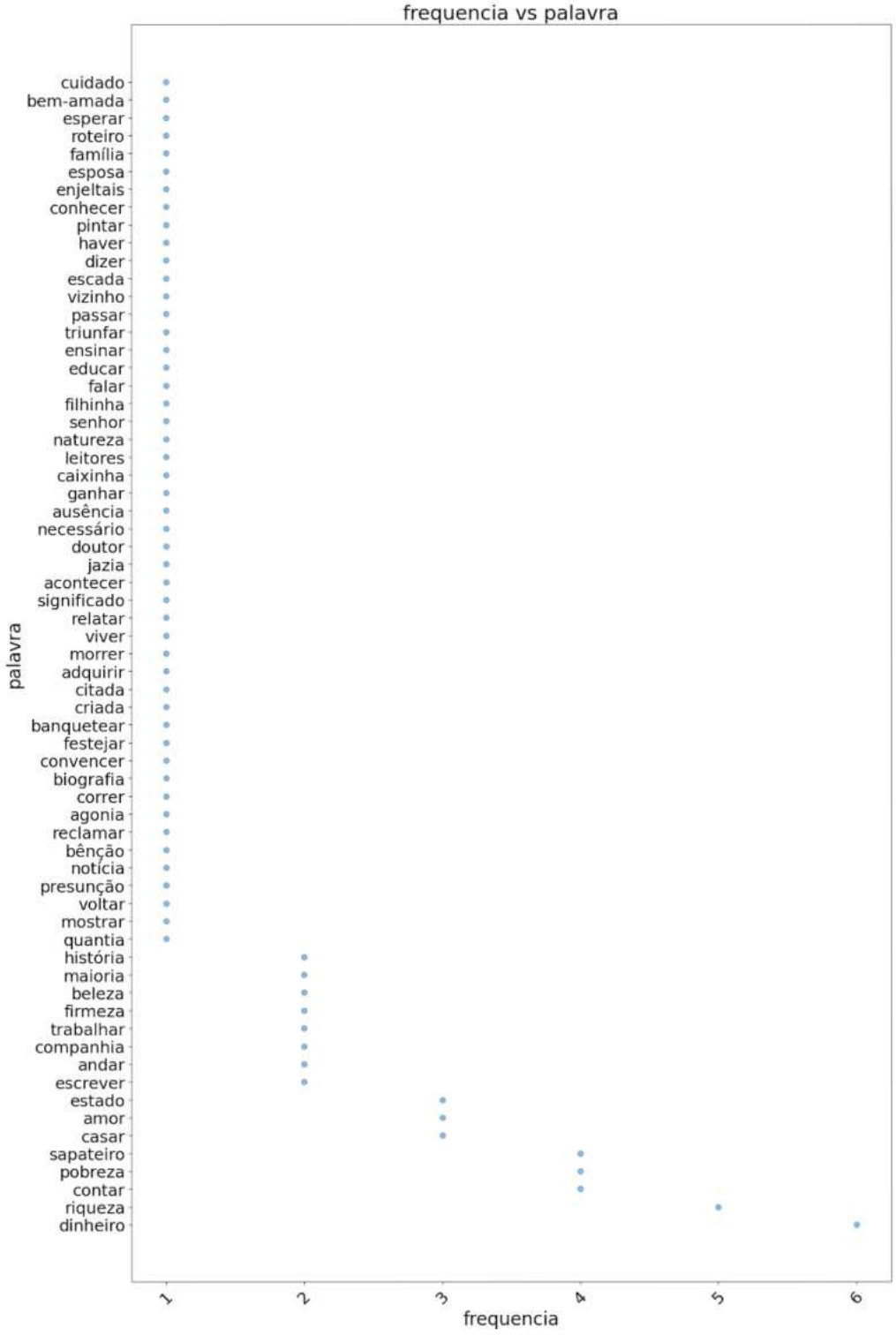
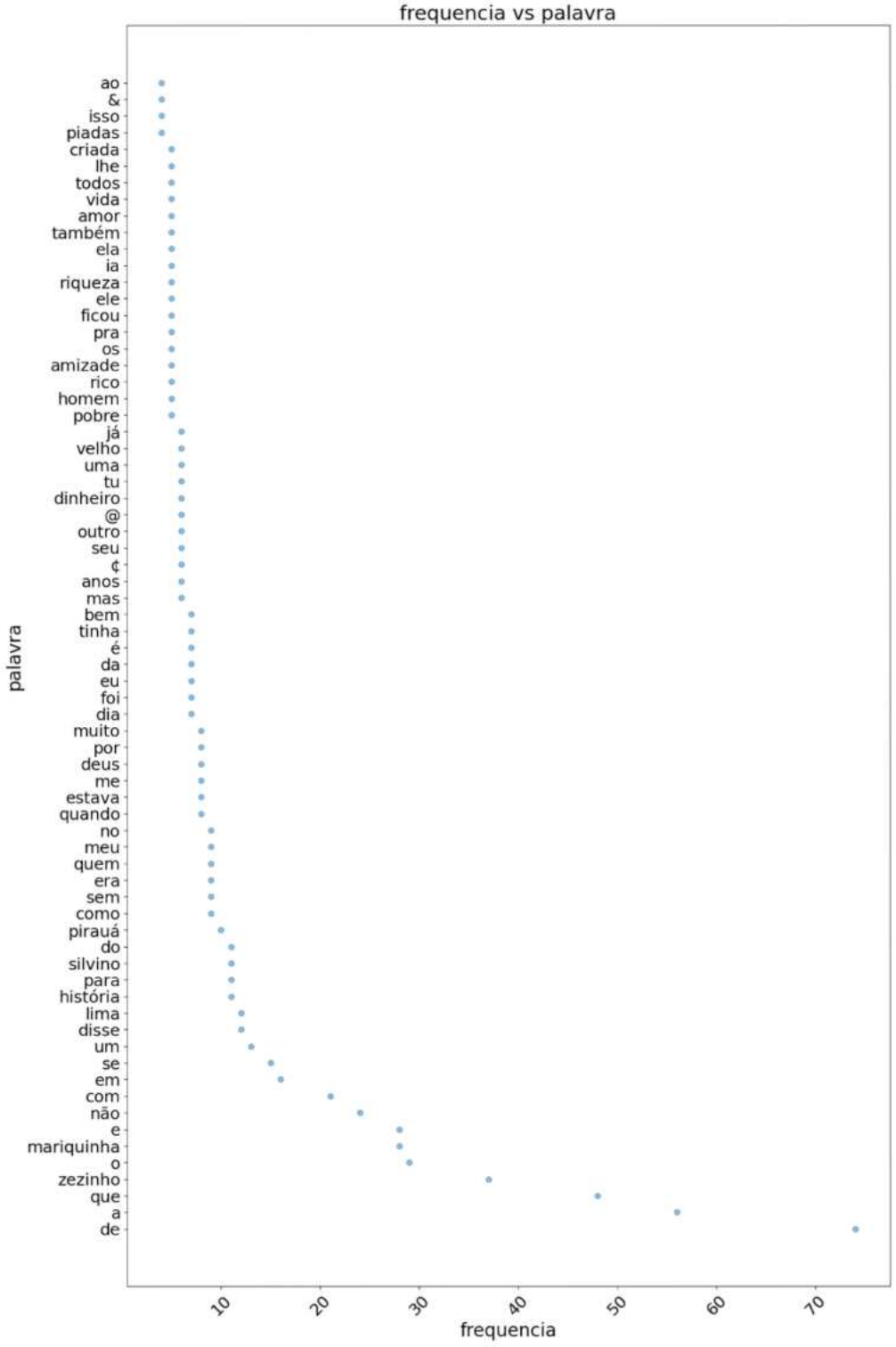


Figura 14: Frequência de ocorrência de todas as palavras - Corpus Cordéis.



5.2.2 Frequência sufixos e radicais

Em relação às perguntas:

- Pergunta 3 - Frequência em que os sufixos aparecem em cada corpus.
- Pergunta 4 - Frequência em que os radicais aparecem em cada corpus.

Podemos notar o mesmo padrão de curva para o corpus de notícias, como mostrado nas Figuras 15 e 16. Este resultado foi gerado utilizando os parâmetros de percentual acumulado de 80% para os sufixos e 3% para os radicais.

Os resultados para os outros dois corpora também foram parecidos com o de notícias, tanto para sufixos quanto para radicais. O resultado para o corpus de livros infantis pode ser visto na Figura 17 e na Figura 18, o do corpus de cordéis na Figura 19 e na Figura 20. A Tabela 6 mostra os parâmetros que foram utilizados para gerar estes mesmos resultados, estes parâmetros se referem a quantidade de palavras que totalizam X% do corpus. Por exemplo, o resultado do gráfico de sufixo para o corpus de notícias foi gerado utilizando as palavras que totalizam 80% do corpus.

Tabela 6: Parâmetros utilizados em cada corpus.

CORPUS	PARÂMETRO SUFIXOS	PARÂMETRO RADICAIS
Notícias	80%	3%
Livros infantis	80%	6%
Cordéis	100%	100%

Figura 15: Frequência de sufixos - Corpus Notícias.

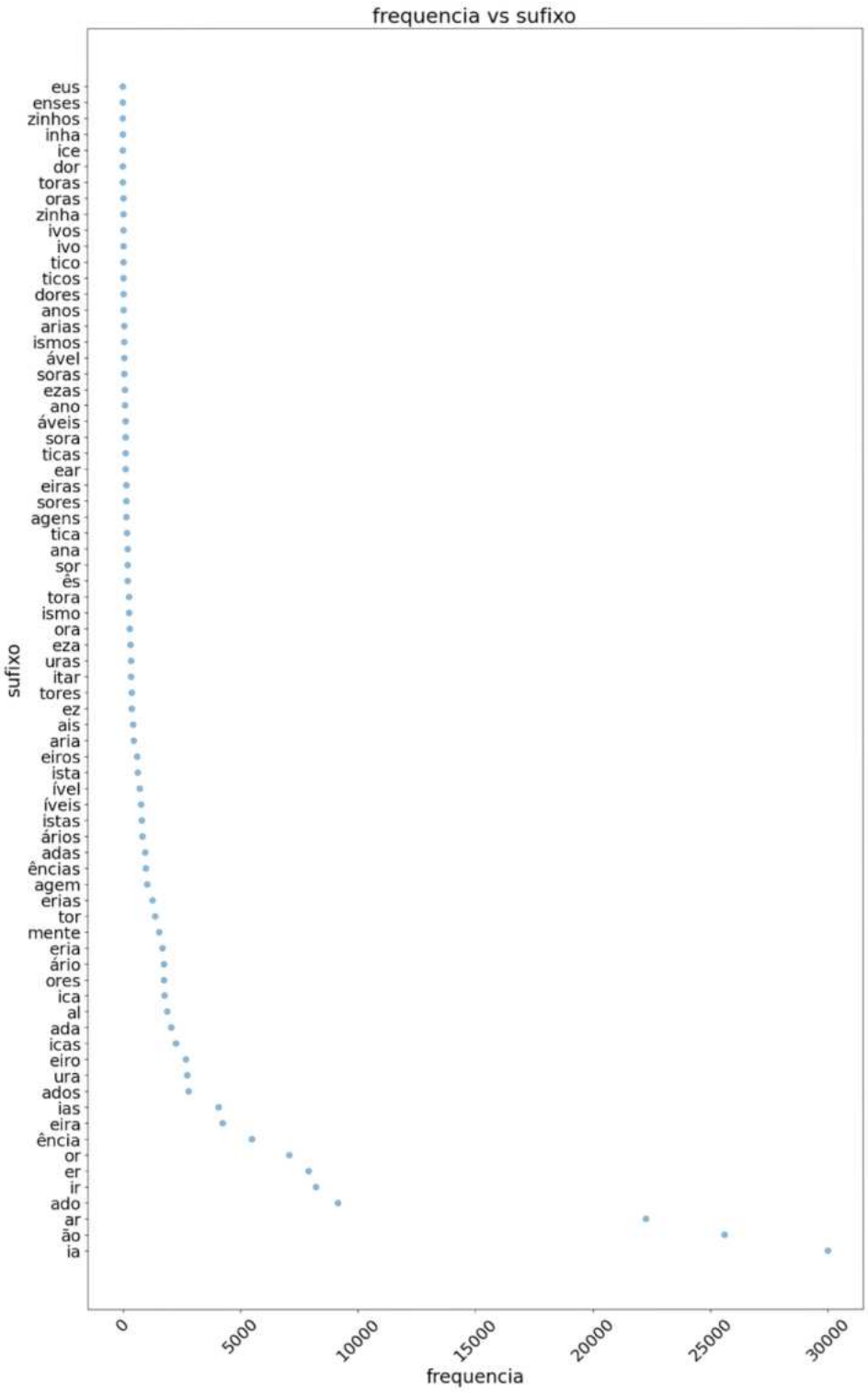


Figura 16: Frequência de radicais - Corpus Notícias.

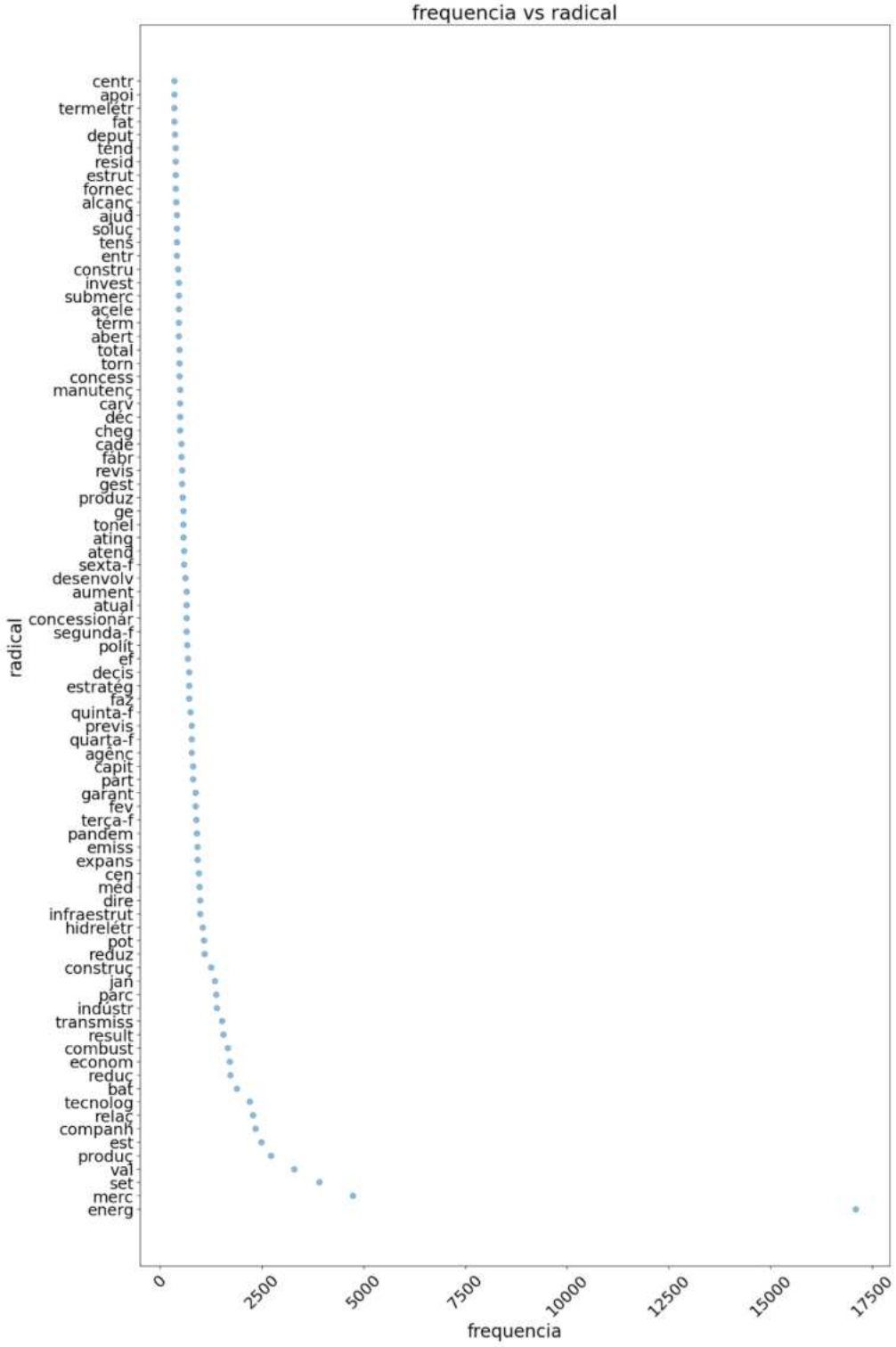


Figura 17: Frequência de sufixos - Corpus Livros Infantis.

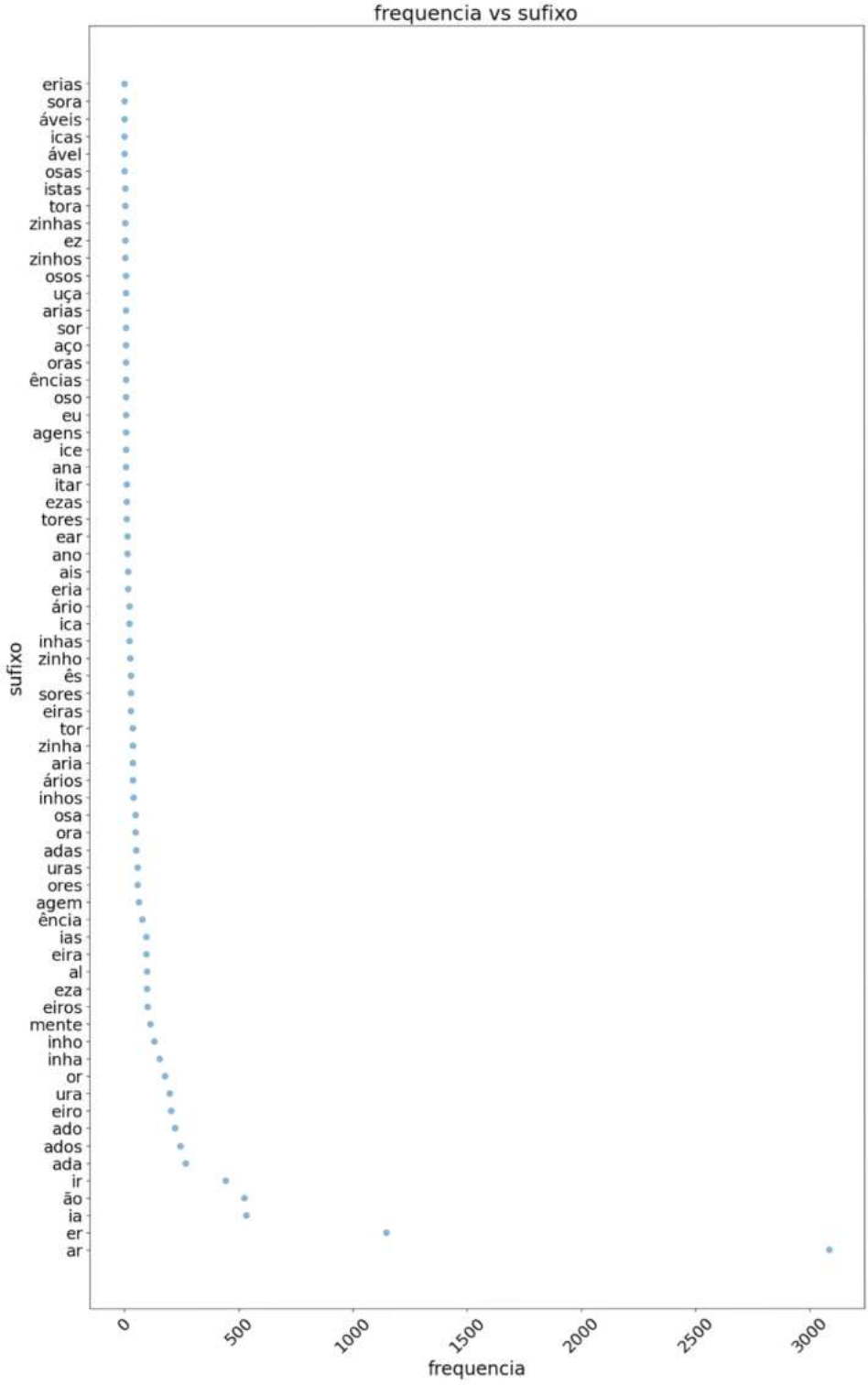


Figura 18: Frequência de radicais - Corpus Livros Infantis.

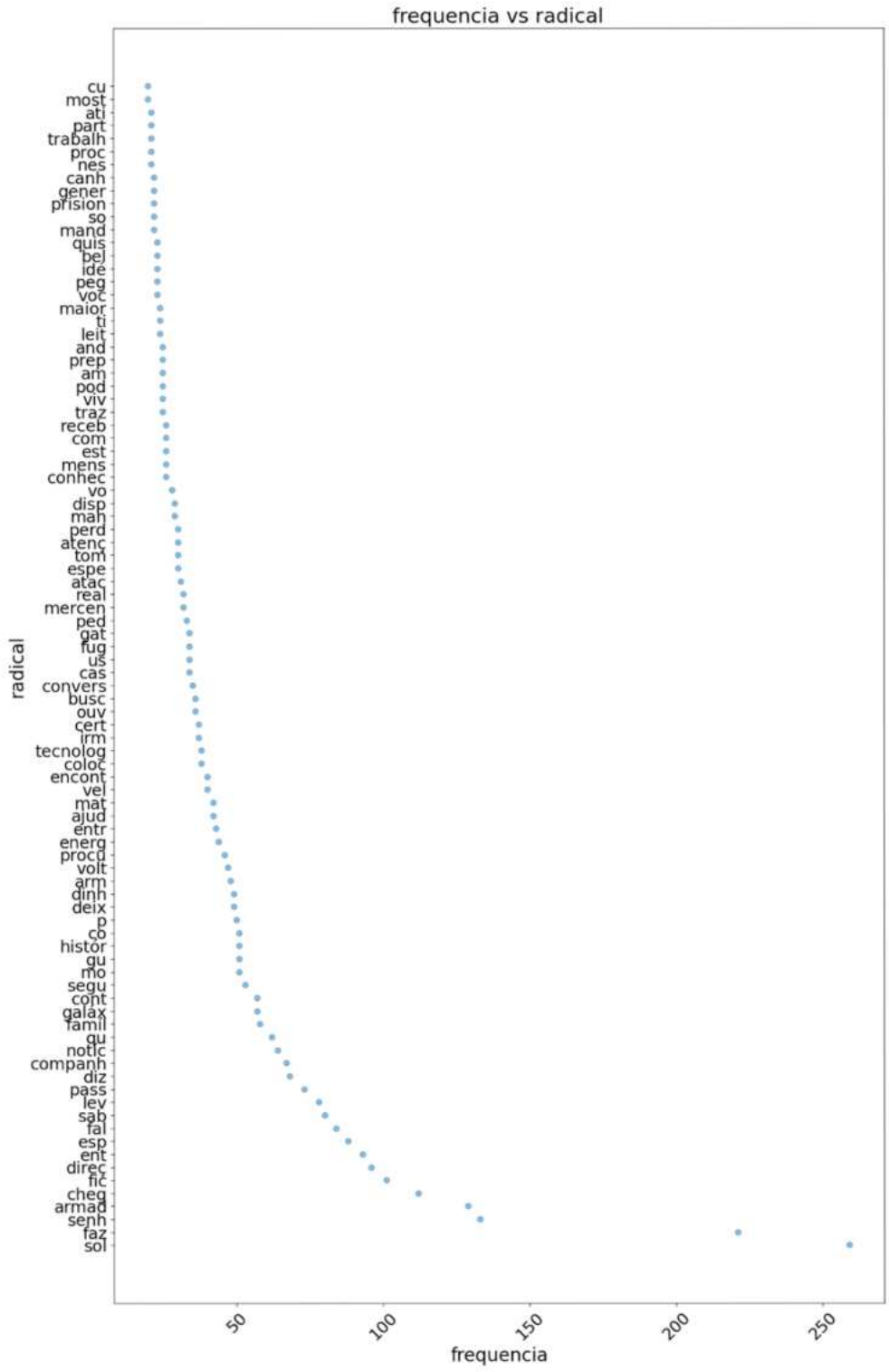


Figura 19: Frequência de sufixos - Corpus Cordéis.

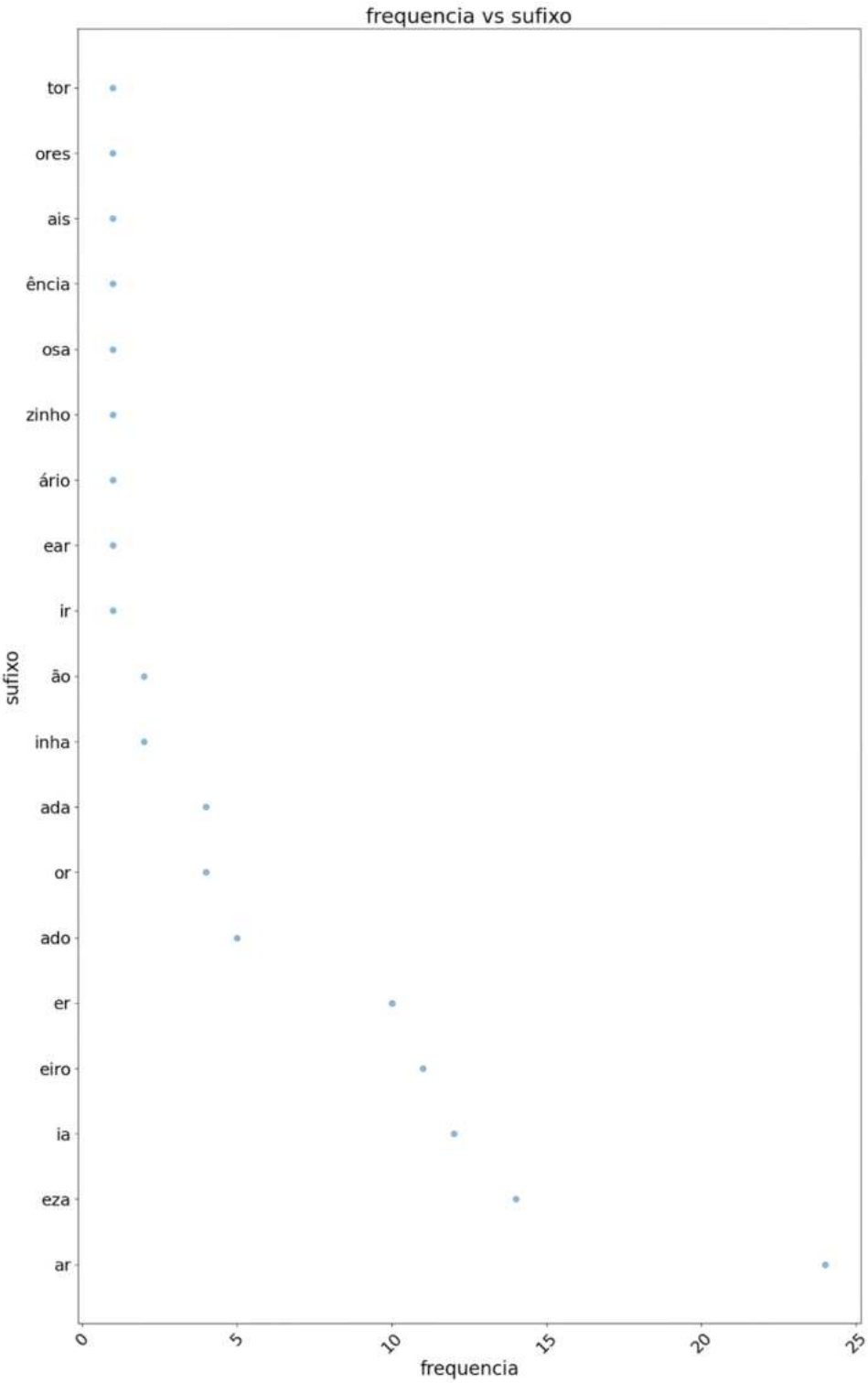
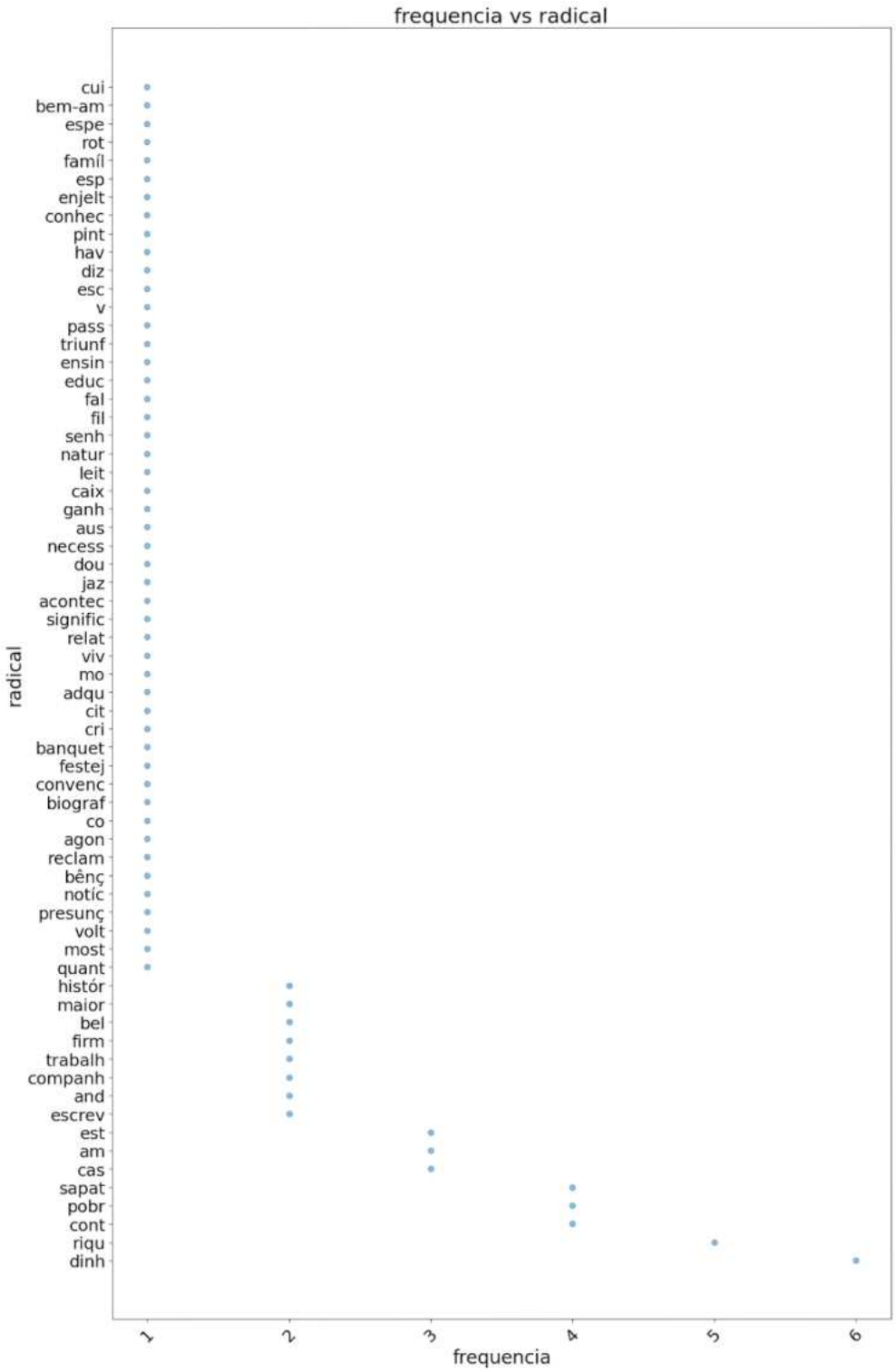


Figura 20: Frequência de radicais - Corpus Cordéis.



Além destas figuras apresentadas, foi possível obter a frequência dos sufixos que não ocorreram em cada corpus. Esta relação é apresentada na Tabela 7.

Tabela 7: Relação de ocorrência de sufixos por corpus.

CORPUS	QTD. TOTAL DE SUFIXOS VÁLIDOS	QTD. DE SUFIXOS COM OCORRÊNCIA	QTD. DE SUFIXOS SEM OCORRÊNCIA	% DE NÃO OCORRÊNCIA
Notícias	247	93	154	~ 62%
Livros infantis	247	86	161	~ 65%
Cordéis	247	19	228	~ 92%

Os resultados da Tabela 7 estão de acordo com o esperado, visto que o corpus com mais palavras distintas deveria apresentar a maior quantidade de ocorrência de sufixos, enquanto o corpus com menor quantidade de palavras distintas deveria apresentar a menor quantidade destas ocorrências. Isto é comprovado nos resultados da tabela.

5.2.3 Quantidade de radicais por classe

As Tabelas 9, 10 e 11, a seguir, mostram os resultados obtidos para a pergunta:

- Pergunta 5 - Quantidade de radicais por classe em cada corpus.

Pôde-se observar uma relação de crescimento acelerado na ocorrência de radicais da classe gramatical dos verbos (VERB) em comparação com as demais classes, quando levado em consideração os valores obtidos em cada corpus. Além disso, é possível dizer também que os verbos e os advérbios possuem um papel mais significante que os substantivos quando se trata do aprendizado da língua portuguesa, pois para iniciantes na língua, o seu entendimento se torna mais rápido quando iniciado pelos verbos e advérbios, por serem utilizados mais vezes que as outras classes.

O corpus de cordéis saiu bastante do padrão, quando comparado com os dois anteriores, porém isso pode ser consequência da diferença na quantidade de palavras que este possui, que é muito abaixo dos demais. É possível notar que não houve nenhum radical da classe ADV, mas que isso pode ser consequência deste mesmo problema descrito.

Tabela 8: Ocorrência de radicais por classe gramatical, utilizando 100% das palavras - Corpus Notícias

CLASSE GRAMATICAL	OCORRÊNCIA	% OCORRÊNCIA
VERB	122.729	~ 75%
ADV	38.877	~ 24%
NOUN	1.553	~ 1%

Tabela 9: Ocorrência de radicais por classe gramatical, utilizando 100% das palavras - Corpus Livros Infantis

CLASSE GRAMATICAL	OCORRÊNCIA	% OCORRÊNCIA[12]
ADV	4.697	~ 52%
VERB	4.151	~ 46%
NOUN	114	~ 1%

Tabela 10: Ocorrência de radicais por classe gramatical, utilizando 100% das palavras - Corpus Cordéis

CLASSE GRAMATICAL	OCORRÊNCIA	% OCORRÊNCIA
NOUN	61	~ 63%
VERB	36	~ 37%
ADV	0	0%

5.2.4 Percentuais

As perguntas abaixo são respondidas na Tabela 11:

- Pergunta 6 - Razão entre o total de palavras com sufixo e o total de palavras por corpus.
- Pergunta 7 - Dentre as palavras que podem possuir sufixos (verbos, advérbios e substantivos), quantas de fato possuem um sufixo.

Os três corpora apresentam o mesmo padrão de respostas, onde a razão entre o total de palavras com sufixo e o total de palavras das classes selecionadas é maior do que a razão total de palavras com sufixo e o total de palavras.

Um resultado curioso que surge indiretamente é a relação entre os resultados das duas perguntas. Ao calcularmos as razões dos resultados do corpus de notícias e do corpus de livros infantis, observamos que o valor é 2,5 vezes maior em ambos os casos. Esta descoberta é interessante e sugere a possibilidade de uma relação entre as duas perguntas. Apesar disso, teriam que ser feitos testes com diversos corpora com uma quantidade alta de dados para podermos garantir a sua veracidade, pois como dá para notar, no caso dos cordéis essa relação não ocorre.

Tabela 11: Razões percentuais entre sufixos e palavras.

CORPUS	RAZÃO PERGUNTA 6	RAZÃO PERGUNTA 7
Notícias	7,25%	18,46%
Livros infantis	0,4%	1,01%
Cordéis	0,0%	0,01%

[12] Valores não somam 100% devido ao arredondamento.

5.2.5 Quantidade de radicais que possuem sufixos diferentes

Os resultados na Figura 21 e na Figura 22, a seguir, são relativos à última pergunta:

- Pergunta 8 - Quantidade de radicais que possuem sufixos diferentes em cada corpus.

Os resultados obtidos podem ser interessantes sob o ponto de vista dos corpora, pois é possível notar na Figura 21 e na Figura 22 a diferença na ordem das quantidades. No corpus de notícias a variação segue, de certa forma, um movimento decrescente de ocorrência à medida que os radicais aumentam sua quantidade de sufixos. Por outro lado, o corpus de livros infantis não apresentou nenhum padrão visível, assim como o corpus de cordéis, que ocorreram apenas radicais com um único sufixo.

Figura 21: Frequência com que os radicais variam a quantidade de radicais - Corpus Notícias.

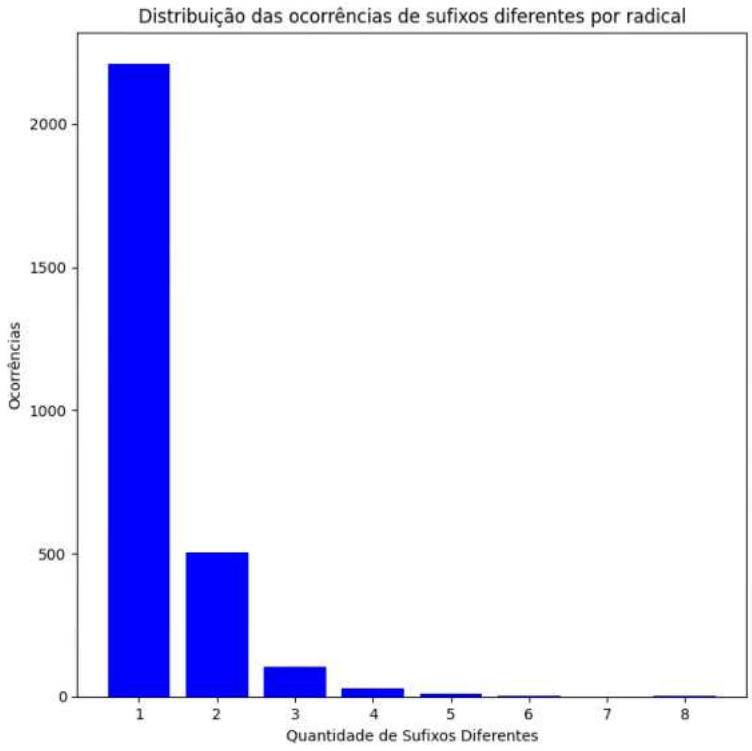
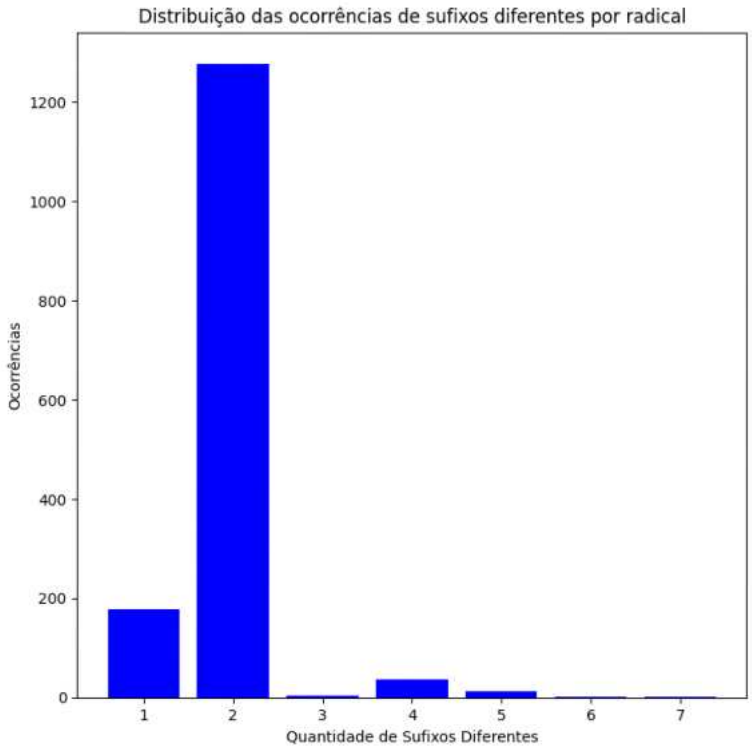


Figura 22: Frequência com que os radicais variam a quantidade de radicais - Corpus Livros Infantis.



O gráfico do corpus dos cordéis não apresentou variações além de um, ou seja, todos os radicais só ocorreram com um único sufixo. Por conta disso, não será apresentado seu gráfico.

A Tabela 12 mostra a quantidade de radicais que aparecem em relação ao total de palavras, ou seja, a quantidade de vezes que um radical aparece sem considerar suas repetições.

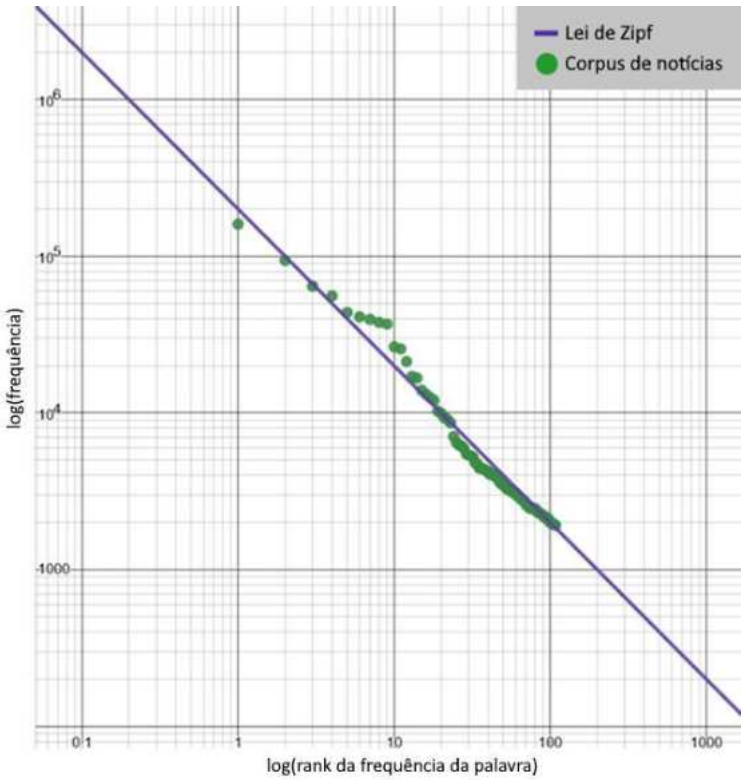
Tabela 12: Contagem de radicais sem repetição.

CORPUS	TOTAL DE PALAVRAS	TOTAL DE RADICAIS ÚNICOS
Notícias	3730	2861
Livros infantis	1835	1511
Cordéis	65	65

5.2.6 Inspeção dos resultados

Para verificar a extração de palavras feita pelo programa, iremos comparar as frequências das palavras obtidas com os valores esperados pela lei de Zipf. Os pontos na Figura 20 exibem a frequência das palavras do corpus de notícias em função da sua posição no ranque de palavras mais comuns. A linha no gráfico denota os valores esperados pela lei de Zipf.

Figura 23: Comparação da Lei de Zipf com o corpus de notícias.



Pela Figura 23 é possível notar que a frequência das palavras coletadas no corpus, no geral, apresenta uma boa conformidade com a lei de Zipf. Essa relação por si só não é suficiente para atestar a corretude da coleta de palavras, mas é um bom indicador de que os dados coletados estão representando de forma fiel um corpus da língua portuguesa.

Para validar a coleta de sufixos, foi feita uma amostragem das 300 palavras mais comuns do corpus, que foi revisada pela Professora Doutora Daniela Cid de Garcia. No total, foram encontrados 44 erros na amostra. Desses 44, os principais foram:

1. 15 erros referentes sufixo plural -s. Em alguns desses casos o programa sequer identificou o sufixo, em outros, ele extraiu parte do radical da palavra como se fosse o sufixo. Por exemplo, ele identificou o sufixo -erias na palavra “baterias”, sendo que o correto seria apenas -s.
2. 6 erros referentes aos sufixos aumentativos -ão e -ção, onde o sufixo capturado estava errado porque continha parte do radical. Esse tipo de erro já era esperado considerando que o sufixo -ão está contido em -ção, o que torna a distinção mais difícil para o algoritmo.

3. 6 erros referentes ao sufixo -ico/-ica. O programa não identificou nenhuma das ocorrência desse sufixo na amostra.
4. 5 erros referentes ao sufixo -mento. Em todas as ocorrências desse sufixo, o stemmer extraiu uma letra a mais do radical como sendo parte do sufixo. Por exemplo, em "fornecimento", o stemmer identificou o radical "fornec", o que resultaria no sufixo -imento, que não existe.

Os erros restantes eram mais pontuais e distribuídos em vários sufixos. Estão contemplados nessa lista os sufixos -io, -eiro, -ado, -al, -dade, -dor e -es (considerando também as variações no feminino e no plural). Vale notar que, todos esses erros podem ser corrigidos manualmente no banco de dados por meio de requisições SQL simples. Porém, pode não ser factível encontrar todas as ocorrências de algum determinado erro na base de dados.

6 CONCLUSÃO

Uma ferramenta de processamento de linguagem natural em português com foco na extração de sufixos se mostra útil em aplicações de análise linguísticas. Ao extrair sufixos das palavras em um texto, a ferramenta pode ajudar na identificação de padrões morfológicos, permitindo uma análise mais profunda da estrutura das palavras e suas relações gramaticais.

Pelo módulo de criação de corpora, foi criado um corpus de livros infantis e um de cordéis, ambos processados a partir de textos contidos em imagens e em arquivos PDF, resultando em corpora estruturados. A ferramenta se mostrou valiosa dada sua versatilidade e aplicação em diversos projetos que necessitam de um corpus. Os resultados demonstram a eficácia e a utilidade do módulo, que não se limita a ser utilizado exclusivamente com os outros módulos apresentados neste trabalho, mas pode ser integrado a qualquer projeto, ampliando seu alcance e utilidade.

A partir do módulo de processamento, foi possível criar um banco de dados contendo informações morfológicas dos corpora, incluindo o resultado da coleta de sufixos de cada token de cada frase. A estruturação e armazenamento dos dados dos corpora em um banco de dados facilitou bastante a realização das análises e o processamento de cada corpus, além de viabilizar o uso desses dados em estudos futuros. Manter todos os dados do processamento na memória do sistema se mostrou muito ineficiente, tanto pelo consumo de recursos computacionais quanto pela organização e modelagem dos dados.

Assim como os anteriores, o módulo de análises pode ser adaptado para fornecer análises para outros formatos estruturados de dados e não apenas para o banco de dados desenvolvido. Essa abordagem destaca a versatilidade e a utilidade do trabalho, apresentando um conceito de "três em um" (três trabalhos em um), onde cada módulo pode ser usado de forma independente ou em conjunto para diversos fins.

Os resultados obtidos apresentaram-se positivos de maneira geral. Uma observação relevante que surgiu é a relação entre a qualidade do texto de entrada e a eficiência do sistema de classificação de livros infantis. Verificou-se que textos submetidos a um tratamento mais refinado contribuem para resultados mais precisos, que representam melhor a morfologia da linguagem. Tal constatação enfatiza a importância do tratamento adequado dos dados textuais, destacando que a qualidade inicial do texto exerce influência direta sobre a qualidade do resultado.

Além disso, foi notável que a curva de ocorrência de palavras seguiu um padrão esperado pela Lei de Zipf, que define um comportamento comum em linguagens naturais, o que é um potencial indicador da precisão e robustez da ferramenta desenvolvida. Em relação a análise realizada, foi possível notar que a ferramenta é capaz de gerar informações estatísticas sobre o uso da linguagem, como evidenciado pelas razões percentuais da subseção 5.2.4, reforçando seu potencial como recurso analítico para estudos linguísticos e educacionais. Esses resultados corroboram a eficácia do programa e fornecem uma base sólida para futuras melhorias e aplicações.

A proposta inicial era de criarmos uma ferramenta classificadora de livros infantis baseada na complexidade do seu texto a partir de dados morfológicos, com o intuito de facilitar a escolha dos livros propostos por faixa etária. Já existe um sistema de classificação etária para diversas mídias no Brasil, o Classind[13], mas não existe um sistema oficial para livros. A ideia seria avaliar analiticamente se um livro é apropriado ou não para uma determinada idade. Porém, não foi possível seguir por este caminho, visto que não havia dados suficientes para treinar um modelo de classificação. Por conta disso, foi necessário coletar e criar uma base de dados morfológicos dos textos, o que permitiria a criação da ferramenta de classificação em um momento futuro.

Como os dados coletados pelo programa precisam ser revisados por uma autoridade, e dificilmente um profissional da área de letras terá experiência com bancos de dados, seria interessante desenvolver uma interface amigável para o usuário interagir e manipular os dados coletados pela ferramenta. Inclusive, durante o desenvolvimento, foi necessário exportar e estruturar dados do banco para que sejam revisados, o que demandou tempo. Uma interface gráfica iria permitir qualquer usuário leigo coletar os dados que precisa, dispensando a necessidade de exportar e formatar os dados manualmente. Este trabalho teve ênfase na extração e análise de sufixos, porém os dados coletados viabilizam estudos focados em outras vertentes, tais como a extração e análise de prefixos, dissidências, vogais temáticas etc.

[13] https://www.gov.br/mj/pt-br/assuntos/seus-direitos/classificacao-1

REFERÊNCIAS

ABAURRE, M. L.; PONTARA, M. N. **Gramática. Texto. Análise e Construção de Sentido**. 1. ed. [S.l.]: Moderna, 2006.

BASÍLIO, M. **Teoria Lexical**. São Paulo: Atica S.A., 1987.

CHICHE, A.; YITAGESU, B. Part of speech tagging: a systematic review of deep learning and machine learning approaches. **Journal of Big Data**, v. 9, n. 1, p. 10, jan. 2022. ISSN 2196-1115.

CUNHA, C.; CINTRA, L. **Nova gramática do português contemporâneo**. 7ª. ed. Rio de Janeiro: Lexikon Editora Digital, 2016.

DE QUADROS, R. M.; STUMPF, M. R.; LEITE, T. D. A. **Estudos da Língua Brasileira de Sinais**. Florianópolis, SC: Editora Insular, v. 1, 2013.

FRAKES, W.; FOX, C. Strength and similarity of affix removal stemming algorithms. **ACM SIGIR Forum**, v. 37, n. 1, p. 26-30, abr. 2003. ISSN 10.1145/945546.945548.

GONÇALVES, C. A. **Morfologia**. 1. ed. São Paulo: Parábola Editorial, 2019.

LIU, J. *et al.* Design and Construction of a Knowledge Database for Learning Japanese Grammar Using Natural Language Processing and Machine Learning Techniques. **2022 4th International Conference on Natural Language Processing (ICNLP)**, Xi'an, China, 19 set. 2022., p. 371-375

MENESES, U. T. B. D. A literatura de cordel como patrimônio cultural. **Revista do Instituto de Estudos Brasileiros**, São Paulo, v. 72, p. 225-244, abr. 2019. ISSN 10.11606/issn.2316-901X.v0i72p225-244.

MORENO-SÁNCHEZ, I.; CORRAL, Á. Large-Scale Analysis of Zipf's Law in English Texts. **PloS one**, Barcelona, v. XI, n. 1, Janeiro. 2016. ISSN 1932-6203.

OLIVEIRA, T. D. N. **Construção e classificação de uma base textual em português**. Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2023.

RAMADITIYA, A. *et al.* Implementation Chatbot Whatsapp using Python Programming for Broadcast and Reply Message Automatically. **2021 International Symposium on Electronics and Smart Devices (ISESD)**, Bandung, Indonesia, 29-30 Junho 2021., p. 1-4

RASTLE, K. The place of morphology in learning to read in. **Cortex**, Londres, v. 116, p. 45-54, 2019. ISSN 0010-9452.

SMITH, R. W. History of the Tesseract OCR engine: what worked and what didn't. **In:** ZANIBBI, R.; COÜASNON, B. **Document Recognition and Retrieval XX**. Mountain View: [s.n.], v. 8658, 2013., p. 865802.

ZIPF, K. G. **The Psycho-Biology of Language**. Londres: George Routledge & Sons, Ltd., v. XXI, 1936.

ZONG, Z.; HONG, C. On Application of Natural Language Processing in Machine Translation. **2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)**, Huhhot, China, 14-16 Setembro 2018., p. 506-510

GLOSSÁRIO

Pseudoprefixo – Afixo no início de uma palavra que pode ser confundido com um prefixo. Exemplos: des-te, re-zar, re-mar.

Pseudosufixo – Afixo no final de uma palavra que pode ser confundido com um sufixo. Exemplos: Jan-eiro, Coraç-ão, m-acho.

Part-of-speech – Parte-da-fala em tradução livre. Uma categorização que agrupa itens léxicos que possuem propriedades gramaticais semelhantes.

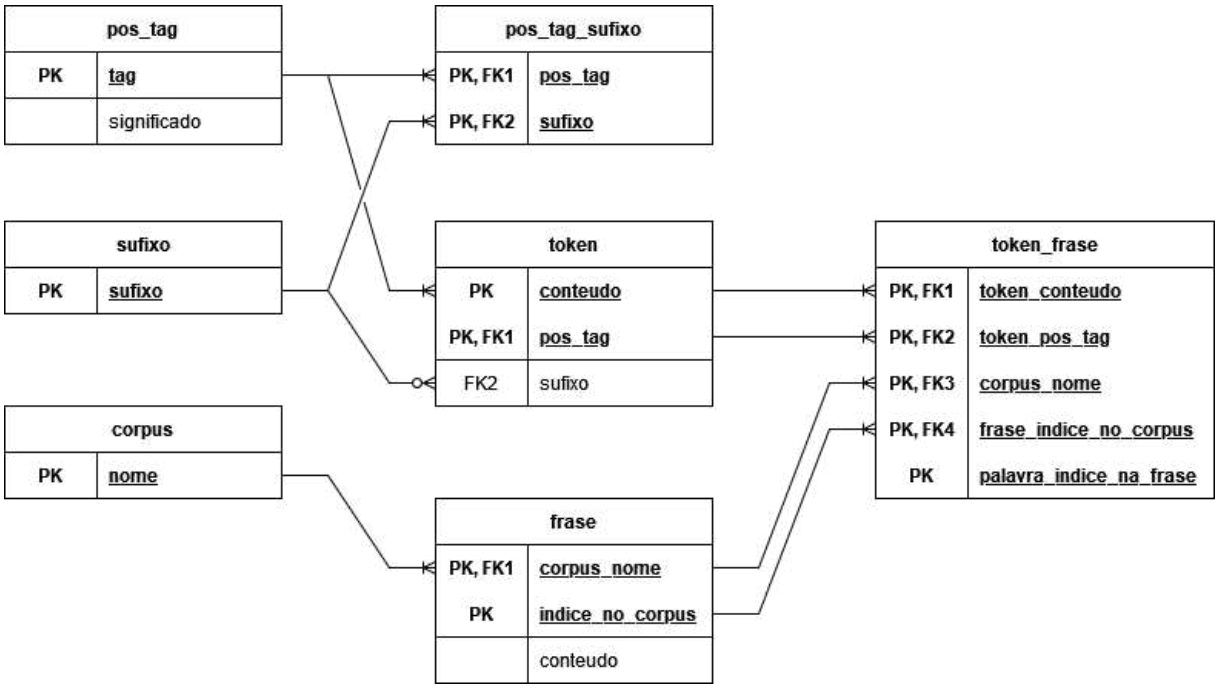
Stopwords - São palavras frequentemente usadas que são filtradas em análises de texto por terem pouco significado analítico, como preposições e artigos.

APÊNDICES

APÊNDICE A – LISTA COMPLETA DE SUFIXOS

CLASSE	SUFIXOS
Nominal / Noun	‘aca’, ‘acas’, ‘acho’, ‘achos’, ‘aco’, ‘aço’, ‘acos’, ‘aços’, ‘ada’, ‘adas’, ‘ado’, ‘ados’, ‘agem’, ‘agens’, ‘aica’, ‘aicas’, ‘aico’, ‘aicos’, ‘ais’, ‘al’, ‘alhão’, ‘alhões’, ‘alhona’, ‘alhonas’, ‘ama’, ‘amas’, ‘ame’, ‘ames’, ‘ana’, ‘anas’, ‘ança’, ‘anças’, ‘ano’, ‘anos’, ‘ão’, ‘aréis’, ‘aréu’, ‘aria’, ‘ária’, ‘arias’, ‘árias’, ‘ário’, ‘ários’, ‘arra’, ‘arras’, ‘ata’, ‘atas’, ‘ato’, ‘atos’, ‘áveis’, ‘ável’, ‘az’, ‘azes’, ‘ção’, ‘ções’, ‘dade’, ‘dades’, ‘dor’, ‘dora’, ‘doras’, ‘dores’, ‘douro’, ‘douros’, ‘ea’, ‘eas’, ‘eca’, ‘ecas’, ‘eco’, ‘ecos’, ‘edo’, ‘edos’, ‘éia’, ‘éias’, ‘eira’, ‘eirão’, ‘eiras’, ‘eiro’, ‘eirões’, ‘eirona’, ‘eironas’, ‘eiros’, ‘ela’, ‘elas’, ‘ena’, ‘enas’, ‘ença’, ‘enças’, ‘ência’, ‘ências’, ‘enha’, ‘enhas’, ‘enho’, ‘enhos’, ‘eno’, ‘enos’, ‘ense’, ‘enses’, ‘enta’, ‘entas’, ‘ento’, ‘entos’, ‘eo’, ‘eos’, ‘eria’, ‘erias’, ‘ês’, ‘êsa’, ‘êsas’, ‘êses’, ‘eu’, ‘eus’, ‘ez’, ‘eza’, ‘ezas’, ‘ezes’, ‘ia’, ‘íaca’, ‘íacas’, ‘íaco’, ‘íacos’, ‘ias’, ‘ica’, ‘iça’, ‘icas’, ‘iças’, ‘ice’, ‘ices’, ‘icha’, ‘ichas’, ‘icho’, ‘ichos’, ‘ície’, ‘ícies’, ‘iço’, ‘iços’, ‘ina’, ‘inas’, ‘inha’, ‘inhas’, ‘inho’, ‘inhos’, ‘ino’, ‘inos’, ‘inta’, ‘intas’, ‘into’, ‘intos’, ‘isca’, ‘iscas’, ‘isco’, ‘iscos’, ‘isma’, ‘ismas’, ‘ismo’, ‘ismos’, ‘ista’, ‘istas’, ‘iva’, ‘ivas’, ‘íveis’, ‘ível’, ‘ivo’, ‘ivos’, ‘lenta’, ‘lentas’, ‘lento’, ‘lentos’, ‘menta’, ‘mentas’, ‘mento’, ‘mentos’, ‘nte’, ‘ntes’, ‘ona’, ‘onha’, ‘onhas’, ‘onho’, ‘onhos’, ‘or’, ‘ora’, ‘oras’, ‘ores’, ‘osa’, ‘osas’, ‘oso’, ‘osos’, ‘ota’, ‘otas’, ‘ote’, ‘otes’, ‘óveis’, ‘óvel’, ‘ria’, ‘rias’, ‘rio’, ‘rios’, ‘sor’, ‘sora’, ‘soras’, ‘sores’, ‘tério’, ‘térios’, ‘tica’, ‘ticas’, ‘tico’, ‘ticos’, ‘tor’, ‘tora’, ‘toras’, ‘tores’, ‘tório’, ‘tórios’, ‘uça’, ‘uças’, ‘uço’, ‘uços’, ‘uda’, ‘udas’, ‘ude’, ‘udes’, ‘udo’, ‘udos’, ‘ume’, ‘umes’, ‘ura’, ‘uras’, ‘úveis’, ‘úvel’, ‘zarrão’, ‘zarrões’, ‘zarrona’, ‘zarronas’, ‘zinha’, ‘zinhas’, ‘zinho’, ‘zinhos’
Verbal / Verb	‘ar’, ‘ear’, ‘ear’, ‘ecer’, ‘er’, ‘escer’, ‘icar’, ‘ir’, ‘iscar’, ‘itar’
Adverbial / Adv	‘mente’

APÊNDICE B – DIAGRAMA ER DO BANCO DE DADOS



APÊNDICE C – LIVROS E CORDÉIS PRESENTES NOS CORPORA

Livros Infantis:

- A Bruxa e o Caldeirão - José Leon Machado;
- Amanda e os Nanorobôs - Eliú Quintiliano;
- Chuva e sol – Adelina Lopes Vieira;
- Conto ou não conto – Abel Sidney;
- Dom Quixote – Adelina Lopes Vieira;
- Eu que Vi, eu que Vi (O Resgate dos Animais) - Devison Amorim do Nascimento;
- Histórias da Avózinha - Figueiredo Pimentel;
- Histórias Que Acabam Aqui - Teresa Lopes;
- Meiguice - Adelina Lopes Vieira.

Cordéis:

- A Filha Do Pescador - Leandro Gomes de Barros;
- Força do amor. Alonso e Marina - Leandro Gomes de Barros;
- A Mulher Roubada - Leandro Gomes de Barros;
- A Seca do Ceará - Leandro Gomes de Barros;
- História da Donzela Teodora - Leandro Gomes de Barros;
- Historia da princeza da Pedra Fina - Leandro Gomes de Barros;
- O casamento do bode com a raposa - Firmino Teixeira do Amaral;
- Uma Viagem ao Céu - Leandro Gomes de Barros;
- História de Zezinho e Mariquinha – Silvino Pirauá de Lima.