

ARTHUR AZEVEDO DA SILVA

**UMA ABORDAGEM HÍBRIDA DE CORREÇÃO GRAMATICAL DE
CONTEÚDOS PRODUZIDOS NA LÍNGUA PORTUGUESA**

**SÃO LUÍS - MA
2022**

ARTHUR AZEVEDO DA SILVA

**UMA ABORDAGEM HÍBRIDA DE CORREÇÃO GRAMATICAL DE
CONTEÚDOS PRODUZIDOS NA LÍNGUA PORTUGUESA**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Areolino de Almeida Neto

**SÃO LUÍS - MA
2022**

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Silva, Arthur Azevedo da.

UMA ABORDAGEM HÍBRIDA DE CORREÇÃO GRAMATICAL DE
CONTEÚDOS PRODUZIDOS NA LÍNGUA PORTUGUESA / Arthur
Azevedo da Silva. - São Luís - MA, 2022.

56

Orientador: Prof. Dr. Areolino de Almeida Neto.

.
- Universidade Federal do Maranhão, Centro de
Ciências Exatas e Tecnológicas, Bacharelado em Ciência
da Computação, São Luís - MA, 2022.

1. Processamento de Linguagem Natural. 2. N-grama.
3. Abordagem híbrida. 4. Sujeito-verbo-concordância.
5. Língua portuguesa. I. Neto, Prof. Dr. Areolino de
Almeida . II. , . III. Universidade Federal do Maranhão,
Centro de Ciências Exatas e Tecnológicas, Bacharelado em
Ciência da Computação. IV. Título.

AGRADECIMENTOS

Agradeço a Deus por tudo em minha vida.

A meus pais por todo o suporte, conselhos, ensinamento, exemplo e amor que deram a mim.

A meu irmão Saul e a minha cunhada Sabryna, que são pessoas que eu amo profundamente e a qual nutro um grande respeito e amizade.

A meus avôs Leonílio (in memoriam), Nazilde (in memoriam), Darcy Vêras de Azevedo (in memoriam) e, especialmente, Manoel Pereira de Azevedo, o único avô que pôde acompanhar toda a minha graduação, incentivando-me e apoiando das mais diversas maneiras.

A meus tios Sandra e Marcílio, assim como meu primo Rodrigo, que não dispensaram esforços para me auxiliar desde antes de entrar na faculdade. E a toda a minha família por estarem comigo sempre que precisei.

A minha amada namorada, Taynara Ellen Martins Sousa pelo amor, companhia, preocupação, risadas e companheirismo.

A todos os meus irmãos da Igreja Presbiteriana do Cruzeiro do Anil, por toda a amizade, ensinamento, cuidado e carinho.

Ao meu orientador Areolino de Almeida Neto, pela orientação sábia e segura durante a elaboração deste trabalho de conclusão de curso.

A meus colegas dos grupos 'Cancelados' e 'Codebuilders', pela companhia, momentos de descontração e auxílio durante toda essa jornada.

A Universidade Federal do Maranhão por proporcionar um ensino sério e de qualidade. E a todos os professores e profissionais que desempenham um excelente trabalho e que foram essenciais para o meu aperfeiçoamento durante todos esses anos.

“Não há assunto tão velho que não possa ser dito algo novo sobre ele.” (Fiódor Dostoiévski)

RESUMO

Este trabalho de conclusão de curso tem como objetivo desenvolver um sistema de correção de erros gramaticais (CEG) para a análise e correção de concordância verbal na língua portuguesa. A escrita possibilitou e ainda possibilita uma importante contribuição para o desenvolvimento da civilização humana. A importância de uma escrita correta é fundamental para que estudos, registros e entre outros, sejam benéficos para a sociedade, visto que erros gramaticais podem causar sérios danos ao correto entendimento, eliminando os benefícios obtidos. Porém são bastante custosos os métodos de correção de texto realizados por humanos. Por conta disso, dentro do processamento de linguagem natural (PLN), área que estuda de maneira computacional as linguagens naturais, existe a CEG, que se propõe para detectar e corrigir erros gramaticais. A CEG tipicamente tem três abordagens: estatística, baseada em regras gramaticais e híbrida. A abordagem híbrida, que une o melhor das outras duas abordagens, é uma solução salutar para esse problema. Neste trabalho, a abordagem baseada em regra, sujeito-verbo-concordância (SVC), foi empregada para a análise de erros de concordância verbal e a abordagem estatística n-grama com um classificador de aprendizado de máquina, para a correção dos erros. Este sistema usa três bibliotecas da linguagem python para as tarefas de PLN: spaCy, mlconjug3 e NLTK. A abordagem híbrida mostrou-se eficaz na tarefa de análise de erros, para frases simples, obtendo um *recall* de 94%, uma precisão de 89% e um *f-score* de 91%. Já a correção, mesmo tendo uma precisão abaixo de 50%, acabou tendo um bom desempenho, visto que em línguas como o português, existe mais de uma correção.

Palavras-chave: Correção de Erros Gramaticais, Processamento de Linguagem Naturais, Língua Portuguesa, Abordagem híbrida, Sujeito-Verbo-Objeto, N-grama, Classificador, Aprendizado de Máquina.

ABSTRACT

The present research has the goal of developing a grammatical error correction (GEC) system for the analysis and correction of verbal agreement in the portuguese language. Writing has made and still makes possible to greatly contribute with the development of human civilization. The importance of proper writing is essential for studies, records, among others, that are beneficial to society as a whole, since grammatical errors can cause serious damage to the correct understanding, causing the loss of the benefits already obtained. However, human methods of text correction are very costly. For that reason, within natural language processing (NLP), a field that computationally studies natural languages, there is GEC, which aims to detect and correct grammatical errors. GEC typically has three main approaches: statistical, grammar rule-based, and hybrid. The hybrid approach, which brings together the best of the other two, is a salutary solution to the problem presented. In this research, the rule-based, subject-verb-concordance (SVC) approach was employed for the analysis of verb agreement errors and n-gram statistical approach with a machine learning classifier. This system uses three libraries from python language for the NLP tasks: spaCy, mlconjug3 and NLTK. The hybrid approach proved to be effective in the error analysis task, for simple sentences, obtaining a *recall* of 94%, a precision of 89% and an *f-score* of 91%. The correction, even having a precision below 50%, ended up having a good performance, since in languages like portuguese, there are several correction options..

Keywords: 1. Grammatical Error Correction, Natural Language Processing, Portuguese Language, Hybrid Approach, Subject-Verb-Object, N-gram

LISTA DE ILUSTRAÇÕES

Figura 1 – Tokenização.	16
Figura 2 – Classes Gramaticais.	17
Figura 3 – Árvore de dependência.	19
Figura 4 – Exemplo de uma árvore de sintática através da ferramenta <i>Dependency Viewer</i>	19
Figura 5 – Árvore sintática com sujeito e predicado.	20
Figura 6 – Classificador bidimensional.	23
Figura 7 – Possíveis hiperplanos de separação para um <i>dataset</i> bidimensional e binário.	24
Figura 8 – Uma função parabólica com duas dimensões.	25
Figura 9 – Design do Pipeline do SpaCy.	26
Figura 10 – Exemplo de Design do Pipeline de PLN.	27
Figura 11 – Exemplo de notebook executado com <i>markdown</i> , código e resultados.	29
Figura 12 – Fluxograma do analisador e corretor verbal.	31
Figura 13 – Exemplo de árvore sintática com sujeito composto.	32
Figura 14 – Extração das sentenças das bases de dados do NLTK.	35
Figura 15 – Exemplo de extração do modelo tri-grama.	37
Figura 16 – Exemplo de extração do modelo skip-grama.	37
Figura 17 – <i>Screenshot</i> da saída da pesquisa no modelo n-grama.	38
Figura 18 – <i>Screenshot</i> da saída da pesquisa de um bi-grama no modelo n-grama.	39
Figura 19 – <i>Screenshot</i> da saída da pesquisa no modelo skip-grama.	39
Figura 20 – Bi-gramas vizinhos ao verbo.	40
Figura 21 – Nuvem de palavras.	42
Figura 22 – Nuvem de palavras dos verbos.	42
Figura 23 – Árvore de dependência de 'Eles tinham me falado outra coisa'.	44
Figura 24 – Árvore de dependência de 'Eles tinha me falado outra coisa'.	45

LISTA DE TABELAS

Tabela 1 – Lista de POS <i>Tags</i>	18
Tabela 2 – Exemplos de N-gramas	21
Tabela 3 – Exemplo de Conjugação do <i>mlconjug3</i>	28
Tabela 4 – Exemplo de <i>tokenização</i>	30
Tabela 5 – Exemplos de POS tagging	30
Tabela 6 – Exemplos da análise morfológica do SpaCy	32
Tabela 7 – Exemplos de <i>lematização</i>	33
Tabela 8 – Tabela com a contagem de tri-gramas	38
Tabela 9 – Tabela de probabilidade dos tri-gramas	38
Tabela 10 – Resultados da análise de concordância verbal	43
Tabela 11 – Resultados das correções de erros gramaticais	46

LISTA DE ABREVIATURAS E SIGLAS

CEG	<i>Correção Gramatical de Erros</i>
PLN	<i>Processamento de Linguagem Natural</i>
KMP	<i>Knuth-Morris-Pratt</i>
SVC	<i>Sujeito-Verbo-Concordância</i>
NLTK	<i>Natural Language Toolkit</i>
POS	<i>Part-of-Speech Tagging</i>
API	<i>Application Programming Interface</i>
SVM	<i>Support Vector Machine</i>
SGD	<i>Stochastic Gradient Descent</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO	11
1.1.1	Objetivos específicos	11
1.2	JUSTIFICATIVA	12
1.3	TRABALHOS RELACIONADOS	13
1.4	ESTRUTURA DO TRABALHO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	LINGUAGEM NATURAL	15
2.2	PROCESSAMENTO DE LINGUAGEM NATURAL	15
2.3	CORREÇÃO DE ERROS GRAMATICAIS	20
2.3.1	Abordagem Baseada em Regras	20
2.3.2	Abordagem Estatística	21
2.4	APRENDIZADO DE MÁQUINA	22
2.4.1	Support Vector Machine	22
2.4.2	<i>Stochastic Gradient Descent</i>	23
3	METODOLOGIA	26
3.1	MATERIAIS	26
3.2	MÉTODOS	30
3.2.1	Análise de erros verbais	30
3.2.2	Corretor de Erros Verbais	33
4	RESULTADOS	41
5	CONCLUSÃO	48
5.1	TRABALHOS FUTUROS	48
	REFERÊNCIAS	50

1 INTRODUÇÃO

A comunicação é uma das maiores conquistas da humanidade. Os processos de comunicação suscitaram o interesse das mais diversas ciências. Entre as diversas formas de comunicação, uma que se destaca é a escrita (MATTELART, 2011). Graças aos registros escritos, pode-se estudar e aprender com o passado, evitando repetir os erros cometidos, por exemplo. Desta forma, progredindo nos relacionamentos interpessoais e entre as nações e dentre outras situações.

A escrita materializa os acordos comerciais, as teorias formuladas pelos cientistas, a legislação que regula os comportamentos aceitos e os condenáveis (BRASIL, 1990), além de diversos outros desenvolvimentos alcançados pela humanidade.

A importância da escrita para o desenvolvimento da humanidade é imensurável (NUNES, 2018). Portanto, reconhecendo essa importância, deseja-se com este trabalho, facilitar e possibilitar a correta comunicação escrita durante a utilização do português para a produção de conteúdos escritos.

1.1 Objetivo

O objetivo deste trabalho é a implementação de um sistema para a análise e correção de erros de concordância verbal na língua portuguesa aplicando técnicas de Processamento de Linguagem Natural (PLN). Este sistema é capaz de identificar o sujeito e o verbo em uma frase em português e, se necessário, realizar a correção do verbo.

1.1.1 Objetivos específicos

- Investigar o processo de análise e correção de erros gramaticais (CEG) através de ferramentas de PLN disponíveis para o português;
- Estudar e investigar ferramentas de visualização de dados para PLN.
- Estudar a viabilidade de um sistema para análise e correção de erros de concordância verbal com métodos tradicionais de PLN;
- Implementar um sistema capaz de reconhecer o sujeito e o verbo em uma determinada oração;
- Implementar um sistema capaz de verificar erros de concordância verbal em orações simples e realizar as devidas correções com baixo custo computacional.

1.2 Justificativa

A cada dia com a ajuda dos avanços tecnológicos mais recentes, a quantidade de artigos, trabalhos científicos e inovações, por exemplo, surgem em uma quantidade jamais vista. Atualmente, 4,14 bilhões de pessoas estão conectadas às redes sociais, conforme pesquisa da WeAreSocial em parceria com a Hootsuite (SOUZA, 2020). Por conta disso, o número de conteúdo escrito no meio digital aumentou de maneira substancial.

O mundo virtual exige em sua maior parte, o conhecimento da escrita. Porém muitos sofrem infortúnio por erros intencionais ou por falta de um bom conhecimento gramatical ou ortográfico. Desta forma, causando a eles um prejuízo às suas imagens públicas e profissionais. O jornal The Guardian, por exemplo, noticiou uma série de erros de português que foram cometidos na rede social Twitter, por um ex-membro do governo federal brasileiro (PHILLIPS, 2020).

O papel da Internet está crescendo também como um recurso educacional, usado por milhões de pessoas para aprendizagem, incluindo o de línguas estrangeiras (PERLINA; TSYGANKOVA, 2017). Além disso, de acordo com Kress (2003), ser 'letrado' é um conceito líquido e que exige uma revisão contínua de métodos pedagógicos quando se trata da transição do texto impresso para o meio digital. Tendo essas em questões em mente, as ferramentas de auxílio e correção de texto vem se tornando cada vez mais necessárias.

Como a maioria das pesquisas na área de correção de erros gramaticais (CEG) foca na correção de erros cometidos por alunos de língua inglesa (ROZOVSKAYA; ROTH, 2019), este trabalho visa a abordar erros cometidos na língua portuguesa, especificamente erros de concordância verbal. Pois, de acordo com uma pesquisa da Universidade de Zimbábue, os alunos de português tendem a ter dificuldades na concordância verbal (NHATUVE; CHIPARA, 2017).

Especificamente no Brasil, existe um alto grau de desuso de recursos computacionais por estudantes em idade escolar, entre um dos motivos, está a falta de uma boa infraestrutura de informática disponível (BATISTA; DAMASCENO, 2019). Além disso, o país não conta com uma adequada democratização do acesso a tecnologia de alta performance para o público mais necessitado, mesmo eles já tendo um bom acesso a tecnologia (ALVES, 2021).

Portanto, deve-se compreender que o público que necessita do auxílio ao ensino da concordância verbal tem pouco acesso a tecnologia de ponta. Dito isso, algumas línguas se beneficiam de um número significativo de recursos computacionais, principalmente o inglês, outras não dispõem de tantas ferramentas dessa natureza, entre elas o português (PADOVANI, 2022). Portanto, vê-se necessário estudar formas computacionalmente menos custosas para se lidar com o problema.

Visto a notória dificuldade que estudantes tem com a concordância verbal (NHATUVE; CHIPARA, 2017), este trabalho propõe a criação de uma ferramenta para auxiliar a correção e o seu ensino. Como é salientado em um trabalho sobre educação especial, o professor pode

utilizar diferentes recursos que facilitem a compreensão e o entendimento do aluno (IGISCK *et al.*, 2017).

No caso de línguas morfológicamente ricas e com pouco recursos computacionais, chegam a ter uma precisão semelhante comparando uma rede neural com baseado em regras para tarefas de PLN (WIECHETEK *et al.*, 2021). Portanto, este trabalho utiliza especialmente métodos tradicionais de PLN para a resolução da tarefa de CEG.

1.3 Trabalhos Relacionados

Yeh *et al.* (2017) propõem um detector de erros para o chinês. Nesse trabalho, o n-grama é utilizado para definir a probabilidade de uma determinada sentença e o algoritmo de busca de *strings* de Knuth-Morris-Pratt (KMP), para detecção e correção de erros. Nesse trabalho, também foi utilizado o método n-grama para determinar a probabilidade de sentenças. Esse sistema obteve um desempenho geral de 59%, que é uma média harmônica entre o desempenho da detecção de erros e da correção.

Luna-Ramírez e Jaimez-González (2021) propõem um corretor automático de erros para textos em espanhol utilizando um modelo de linguagem criado a partir do método estatístico n-grama. Esse trabalho se dividiu em três partes: detecção de erros, geração de candidatos para a correção e seleção dos melhores candidatos para correção. A divisão desse trabalho foi a mesma utilizada por esta pesquisa.

Já Huang *et al.* (2019) propõem um método de correção híbrida de erros gramaticais do inglês para alunos chineses, por meio da abordagem estatística e de uma abordagem baseada em regras. Esse trabalho realizou diversos experimentos, entre eles, um chegou a uma taxa de 88% de detecção de erros.

Em Putra e Enda (2020), é proposto um modelo para identificação de erros gramaticais em *software* especificação de requisitos usando estatísticas e técnicas baseadas em regras. Assim como esse trabalho, esta pesquisa construiu um dicionário de frequência n-grama.

Já Boroş *et al.* (2014) propõem um sistema híbrido de CEG para o romeno, utilizando tanto a abordagem estatística, quanto a baseada em regras para a detecção e correção de erros. A CEG desse sistema tem três etapas, a primeira é a correção ortográfica, a segunda é a detecção de erros gramaticais típicos, através da abordagem baseada em regras, e, por fim, o método estatístico para a seleção de correção entre os candidatos gerados. Esse trabalho obteve uma precisão de 31% e um *recall* de 14%.

Em Fahda e Purwarianti (2017), foi proposto um protótipo de verificador ortográfico e gramatical em indonésio que usa uma combinação de regras e métodos estatísticos. O protótipo detecta, corrige e explica erros de ortográficos e gramaticais.

1.4 Estrutura do Trabalho

Incluindo o capítulo presente, este trabalho está dividido em cinco capítulos, sendo que o Capítulo 2 apresenta a fundamentação teórica utilizada como base para o trabalho proposto. Já no Capítulo 3, são descritas as ferramentas e os métodos adotados por este trabalho. No Capítulo 4, são apresentados os resultados obtidos durante esta pesquisa. Por fim, no Capítulo 5, são discutidas as conclusões obtidas neste trabalho de conclusão de curso.

2 FUNDAMENTAÇÃO TEÓRICA

No presente capítulo, a fundamentação teórica para a devida compreensão das técnicas e tecnologias utilizadas é apresentada. O processamento de linguagem natural (PLN), que é a área que a computação que se une com outras para o estudo das linguagens, tem proeminente destaque neste trabalho, assim como a própria língua portuguesa. Além deles, também está presente o conceito de um classificador de aprendizado de máquina.

2.1 Linguagem Natural

Em linguística, neuropsicologia e filosofia da linguagem, uma língua natural é uma língua que tenha evoluído naturalmente no ser humano e na sociedade através do uso e da repetição sem planejamento consciente ou premeditação. As línguas naturais podem assumir diferentes formas, tais como a fala ou a assinatura. Essas línguas distinguem-se das línguas construídas e formais, tais como as utilizadas para programar computadores ou para estudar lógica (LYONS, 1991).

A linguagem natural tem o propósito de possibilitar a comunicação entre as pessoas (FARINON, 2015). No caso do presente trabalho, a língua alvo é o português, idioma esse que, segundo a Comunidade de Países da Língua Portuguesa (ONU, 2021), já é a quinta língua mais falada no mundo, a primeira no hemisfério sul, terá até o final do século mais de 500 milhões de falantes e é uma das línguas mais faladas na internet e nas redes sociais. Dentro da gramática dessa língua, este trabalho se propõe a lidar com a concordância verbal.

A concordância verbal é a conformidade morfológica entre uma classe (neste caso, o verbo) e seu escopo (neste caso, o sujeito). Ela implica, portanto, na redundância de formas, ou seja, se houver marcação de plural no sujeito, haverá marcação de plural no verbo (HUNTER-MANN; CUNHA, 2021). Assim, entende-se que a concordância verbal precisa da harmonia entre sujeito e verbo.

Já Bechara (2012) expõe que a concordância verbal, o sujeito e o verbo concordam em número e pessoa. Segundo essa regra geral, o sujeito composto, ou seja, aquele com dois ou mais núcleos, o verbo estará no plural.

2.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é um campo da ciência da computação, especificamente, da inteligência artificial, responsável por estudar e processar qualquer tipo de manipulação computacional de uma linguagem natural, com o objetivo de construir ferramentas que permitam compreender, analisar e gerar linguagem natural (PINTO, 2015). O PLN trata computacionalmente os diversos aspectos comunicação humana, como sons, palavras, sentenças

e discursos, considerando formatos e referências, estruturas e significados, contextos e usos (GONZALEZ; LIMA, 2003).

Conforme Allen (2003), o PLN refere-se aos sistemas que analisam e buscam compreender línguas humanas, de forma que a entrada do sistema possa ser um texto escrito ou falado e a tarefa possa ser, por exemplo, uma tradução para outro idioma, uma construção de um banco de dados, ou a sumarização do texto. Já Roque *et al.* (2019) expõem que o PLN tem repercussão na compreensão da língua falada ou escrita.

De acordo com Kostareva *et al.* (2016), todo sistema de PLN contém a análise léxica, morfológica, sintática e semântica. As duas primeiras etapas estão relacionadas ao tratamento de palavras, já na terceira, são criadas estruturas que relacionam as palavras e, por fim, na última, é criada uma representação do significado do texto.

Análise Léxica

O léxico é o lugar da estocagem da significação e dos conteúdos significantes da linguagem humana (BIDERMAN, 1996). Portanto, o léxico de um idioma significa a coleção de palavras e seus significados. Conforme Gonzalez e Lima (2003), o termo 'léxico' pode ser utilizado no PLN para identificação de componentes semânticos e gramaticais sobre palavras.

Na etapa da análise léxica, acontece a conversão de uma cadeia de caracteres (o texto da consulta) em uma cadeia de palavras. Assim, o propósito desta etapa é a identificação das palavras que constituem a consulta (GONZALEZ; LIMA, 2003). Então, essa fase é voltada para a criação de *tokens*, que é uma pequena unidade de um texto e podem ser por exemplo, uma palavra, parte de uma palavra, uma pontuação ou um símbolo.

A primeira etapa no PLN é identificar os *tokens*, ou aquelas unidades básicas que não precisam ser decompostas em um processamento subsequente (WEBSTER; KIT, 1992). A *tokenização*, segundo Farinon (2015), é realizada na análise léxica e é uma técnica para decompor um texto em *tokens*, ou seja, separa o texto em palavras, possibilitando a análise morfológica através de sistema de PLN. Um exemplo da *tokenização* pode ser visto na Figura 1.

Figura 1 – Tokenização.

['Que', 'a', 'Força', 'esteja', 'com', 'você']

Fonte: Kenobi (1977)

Análise Morfológica

A análise morfológica é responsável pela análise de palavras isoladas, onde cada palavra é classificada em uma categoria de acordo com as regras que regem a língua portuguesa (FARINON, 2015). Assim, ela analisa a estrutura, a formação e as classificações das palavras, identificando em qual categoria morfológica a palavra *tokenizada* se encaixa – como verbo, substantivo, adjetivo,

entre outros, no caso do português (FORNAZARI, 2021). Ou seja, essa análise é responsável por definir a classe gramatical de um referido *token*. As classes de palavras podem ser vistas na Figura 2.

Figura 2 – Classes Gramaticais.



Fonte: Heredia (2018)

No PLN, assim como na língua portuguesa, o trabalho de classificar as palavras tem o nome de análise morfológica, mas no PLN, as dez classes gramaticais são divididas em mais rótulos, como por exemplo a classe verbo, que tem um rótulo para verbo e outro somente para verbo auxiliar. Essa tarefa também é conhecida como *Part-Of-Speech tagging (POS)*. Na Tabela 1, é possível visualizar cada POS e seu rótulo.

Análise Sintática

Para Luft (2002), a sintaxe é o estudo de regras que regem e constroem uma combinação de palavras para constituir orações. Portanto entende-se que a sintaxe é a área que estuda a relação estrutural entre as palavras que compõem uma oração.

A análise sintática verifica a função de cada termo de uma oração. Um termo, ou palavra, pode ser classificado de forma diferente de acordo com a função que desempenha. Portanto, a análise sintática é o processo de investigação e de identificação da sintaxe das palavras nas orações (SANTANA, 2020). Então, entende-se que é necessário compreender a função de cada

Tabela 1 – Lista de POS *Tags*

POS Tag	Descrição
ADJ	Adjetivos
ADP	Preposição
ADV	Advérbios
AUX	Verbos Auxiliares
CONJ	Conjunções/Conectivos
CCONJ	Conjunções Coordenativas
DET	Artigos
INTJ	Interjeições
NOUN	Substantivos Comuns
NUM	Numerais
PRON	Pronomes
PROPN	Substantivos Próprios
PUNCT	Pontuações
SCONJ	Conjunções Subordinadas
SYM	Simbolos
VERB	Verbos
SPACE	Espaços
X	Outros

Adaptado de (SAKURAI, 2019)

um deles para a realização da análise sintática.

Já no PLN, uma forma de análise sintática é *parsing*, que é o processo que analisa uma sequência de entrada lida de um arquivo de computador ou teclado como uma sentença e determina sua estrutura gramatical (VARGAS; KEPLER, 2012). Então, pode-se entender que o *parsing* é realizado a partir de uma análise de *tokens*.

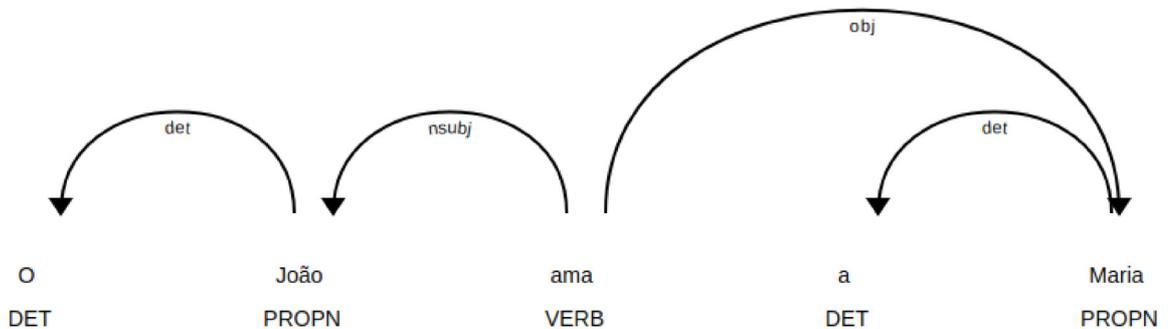
O *parsing* é o procedimento que avalia os vários modos de como combinar regras gramaticais, com a finalidade de gerar uma estrutura de árvore que represente a estrutura sintática da sentença analisada (GONZALEZ; LIMA, 2003). A análise sintática deve explorar os resultados da análise morfológica para construir uma descrição estrutural da frase (RICH *et al.*, 1994). A estrutura criada pela análise sintática pode ser chamada de árvore sintática ou de árvore de dependência.

Conforme Costa (2014), a árvore de dependência obtém informações sobre os relacionamento entre os *tokens*. Portanto, nessa árvore, é possível, por exemplo, reconhecer para qual substantivo determinado verbo se refere e também qual é o núcleo do sujeito, possibilitando uma análise mais apurada do texto. Essa árvore também pode ser chamada de árvore sintática.

Os relacionamento da árvore de dependência, assim como na análise morfológica, são rotulados. Na Figura 3, esses relacionamentos entre os *tokens* estão representados como setas,

que são marcadas como 'det', 'nsubj' e 'obj', que significam respectivamente: artigo, núcleo do sujeito e objeto. Essa imagem foi gerada através da ferramenta Jupyter Notebook.

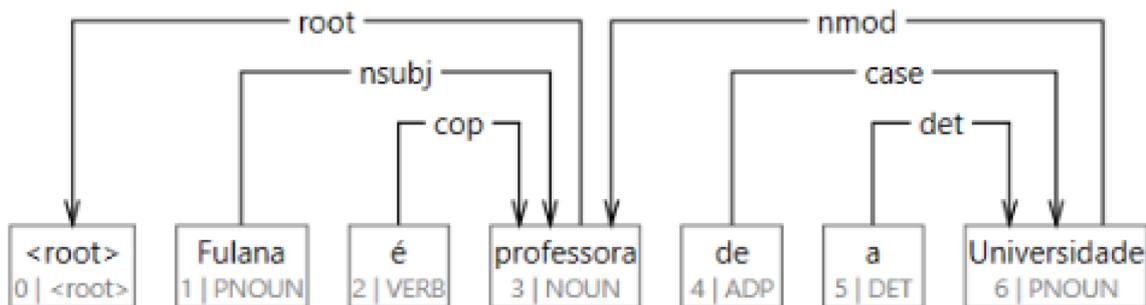
Figura 3 – Árvore de dependência.



Fonte: Do autor

Já Baia *et al.* (2020) expõem que os analisadores de dependência criam estruturas pelas relações gramaticais entre as palavras das sentenças analisadas. Portanto, eles frisam que a árvore de dependência é formada e organizada por regras gramaticais, como é possível observar na Figura 4. Essa figura contém uma árvore criada através da ferramenta denominada *dependency viewer*.

Figura 4 – Exemplo de uma árvore de sintática através da ferramenta *Dependency Viewer*.



Fonte: Baia *et al.* (2020)

Assim, a árvore sintática possibilita não somente obter o relacionamento entre duas palavras, identificando o núcleo do sujeito ou o verbo auxiliar, por exemplo, mas também separando o sujeito do predicado. A Figura 5 apresenta um exemplo do resultado de uma árvore sintática com o seu sujeito e predicado.

Figura 5 – Árvore sintática com sujeito e predicado.



Fonte: Adaptado de Farinon (2015)

2.3 Correção de Erros Gramaticais

A correção de erros gramaticais (CEG) visa a corrigir automaticamente vários tipos de erros em um determinado texto (WANG *et al.*, 2020). Esses erros violam as regras de uma linguagem natural específica. Na tarefa de CEG, pode-se facilmente distinguir duas tarefas separadas: detecção de erros gramaticais e correção desses erros. Tipicamente, existem três tipos de abordagens: baseado em regras, estatístico e híbrido. A dificuldade de detectar e corrigir um erro depende de sua classe (BOROŞ *et al.*, 2014)

2.3.1 Abordagem Baseada em Regras

A abordagem baseada em regras, segundo Shaalan (2010), é uma técnica baseada em sólidos conhecimentos linguísticos, ou seja, utiliza regras definidas e não automaticamente treinadas. Uma das vantagens dessa abordagem é a fácil incorporação de um conhecimento linguístico específico, como uma regra ou exceção, além disso, essas regras podem facilmente ser reutilizadas por outra tarefa de PLN. Ela é recomendada para trabalhos em línguas morfologicamente ricas.

Já Kang *et al.* (2013) comentam que a abordagem baseada em regras oferece diversos benefícios em relação a uma aprendizagem mecânica, por exemplo, principalmente quando se tratando de tarefas como uma das abordadas neste trabalho, que é a detecção e a correção de erros gramaticais.

Uma das grandes vantagens da abordagem baseada em regras, segundo Dorash (2017), é que as regras gramaticais podem ser desenvolvidas de uma maneira bem flexível e que dispõe de uma facilidade para serem atualizadas sem que o sistema sofra com grandes alterações. Portanto, essa técnica possibilita um real conhecimento de como o sistema funciona, inclusive, propiciando uma simplificação também na hora da correção de algum erro ou *bug*.

O SVC (Sujeito-Verbo-Concordância) é uma forma de resolver correções relacionadas ao verbo. De acordo com Xiang *et al.* (2013), esse método é particularmente relacionado ao substantivo do sujeito que é determinado pelo verbo. Esse substantivo pode ser encontrado na

árvore de dependência com o rótulo 'nsubj'.

2.3.2 Abordagem Estatística

As abordagens estatísticas dependem da construção de modelos estatísticos (usando formas de superfície ou rótulos sintáticos) que são usados para detectar e corrigir erros locais. A típica abordagem estatística modela a probabilidade da ocorrência de um evento, dado um histórico de eventos anteriores (BOROŞ *et al.*, 2014). Por conta desse caráter matemático, essa abordagem pode ser adaptada mais facilmente para outra linguagem natural, precisando somente de um conjunto de dados de textos da dessa língua e se necessário, igualmente rotulada.

Uma forma de abordagem estatística é n-grama, que conforme Putra e Enda (2020), consiste no mapeamento de uma sequência de n-itens em valores de probabilidade, onde n representa o número de itens. Usualmente, consiste em uni-grama (um item), bi-grama (dois itens) e tri-grama (três itens). Os itens podem ser palavras, sílabas ou letras dependendo da aplicação. Na Tabela 2, são mostrados alguns exemplos de n-gramas.

Tabela 2 – Exemplos de N-gramas

N-grama	Sentença	Exemplo
Uni-grama	Esse é um exemplo	'Esse', 'é', 'um', 'exemplo'
Bi-grama	Esse é um exemplo	'Esse é', 'é um', 'um exemplo'
Tri-grama	Esse é um exemplo	'Esse é um', 'é um exemplo'

Fonte: Do autor

Segundo Kapadia (2019), a probabilidade de uma tarefa $P(w|h)$, onde w é uma palavra e h é uma história (uma frase ou parte de uma frase), ocorre através da contagem de frequência relativa a partir de um corpus grande. Dessa forma, contam-se quantas vezes a história aparece no corpus e o número de vezes que a palavra aparece ligada a essa história. O primeiro conhecimento importante para a solução disto é que as probabilidades dos n-gramas podem ser obtidas a partir das frequências esperadas associadas para n-grama e (n-1) gramas (STOLCKE; SEGAL, 1994):

$$P(w_n|w_1, w_2 \dots w_{n-1}) = \frac{C(w_1 \dots w_n | L)}{C(w_1 \dots w_{n-1} | L)} \quad (2.1)$$

onde $c(w|L)$ representa a contagem esperada de ocorrências de uma sequência w em uma sentença de L .

De acordo com Putra e Enda (2020), para calcular o valor de probabilidade de cada n-grama em uma frase, pode ser usada a equação de estimativa de máxima verossimilhança. Nas equações a seguir, são mostrados como os modelos de bi-grama e tri-grama aproximam a probabilidade de uma palavra dadas todas as palavras anteriores usando apenas a probabilidade condicional de uma palavra precedente:

$$P(w_i|w_{i-1}) = \frac{N(w_{i-1}, w_i)}{N(w_{i-1})} \quad (2.2)$$

$$P(w_i | w_{i-2}, w_{i-1}) = \frac{N(w_{i-2}, w_{i-1}, w_i)}{N(w_{i-2}, w_{i-1})} \quad (2.3)$$

onde $N(w_{i-2}, w_{i-1}, w_i)$ e $N(w_{i-2}, w_{i-1})$, por exemplo indicam o número de ocorrências de uma sequência $N(w_{i-2}, w_{i-1}, w_i)$ em um determinado corpus.

Através da equação de estimativa de máxima verossimilhança, o cálculo da probabilidade da sentença 'Eu tenho um' seja seguida por 'sonho' é feito da seguinte maneira:

$$P(\text{sonho} | [\text{eu}, \text{tenho}, \text{um}]) = \frac{C([\text{eu}, \text{tenho}, \text{um}, \text{sonho}])}{C([\text{eu}, \text{tenho}, \text{um}])} \quad (2.4)$$

onde $C[\text{eu}, \text{tenho}, \text{um}, \text{sonho}]$ representa o total de ocorrências da sequência 'eu tenho um sonho' no corpus, e, $C[\text{eu}, \text{tenho}, \text{um}]$, o total de vezes que 'eu tenho um' aparece no corpus.

Portanto o n-grama pode ser usado para a criação de um modelo linguístico. O propósito de um modelo linguístico é fornecer a probabilidade da ocorrência de uma palavra aparecer em determinado contexto (ŠOIC *et al.*, 2021). Geralmente, um modelo n-grama é construído a partir de um grande corpus linguístico, do qual a sequência de palavras é extraída. Como resultado, o modelo contém informações sobre transições de palavras e de como elas ocorrem no uso da linguagem específica (ŠOIC *et al.*, 2021).

2.4 Aprendizado de Máquina

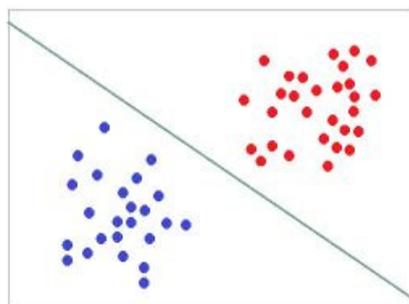
O aprendizado de máquina é um subcampo da inteligência artificial que tem como propósito a elaboração de sistemas capazes de aprender automaticamente (HOSCH, 2021). Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores (MONARD; BARANAUSKAS, 2003). Diferentes técnicas de aprendizagem, ou combinações delas, têm sido usadas para análise de dados textuais em diversos contextos (BRUNIALTI *et al.*, 2015).

No aprendizado de máquinas, um classificador é responsável por dizer a qual classe determinada instância deve pertencer. Já um classificador linear tem como objetivo encontrar uma função linear capaz de separar as amostras em suas respectivas classes, de acordo com as suas características, construindo assim, o classificador (SOUZA, 2018). A Figura 6 contém um exemplo de classificação linear em uma base de dados bidimensional, onde uma reta separa as amostras. *Stochastic gradient descent* (SGD) e o *support vector machine* (SVC) são exemplos de classificadores lineares.

2.4.1 Support Vector Machine

A máquina de vetor de suporte (do inglês *Support Vector Machine* (SVM)) foi desenvolvida por Vapnick (1998) para uma classificação binária. O SVM é embasado pela teoria de aprendizado estatístico que estabelece uma série de princípios que devem ser seguidos na

Figura 6 – Classificador bidimensional.



Fonte: Souza (2018)

obtenção de classificadores com boa generalização (LORENA; CARVALHO, 2007). Entre esses princípios estão a robustez em grandes dimensões, boa capacidade de generalização e a convexidade da função objetivo (GONÇALVES, 2010).

A capacidade de generalização de um classificador é medida por sua eficiência na classificação de dados que não pertencem ao conjunto utilizado em seu treinamento (GONÇALVES, 2010). As SVMs são robustas diante de objetos de grandes dimensões, por exemplo, em imagens (FERRÃO *et al.*, 2007). As SVMs implica a otimização de uma função quadrática, que possui apenas um mínimo global (GONÇALVES, 2010).

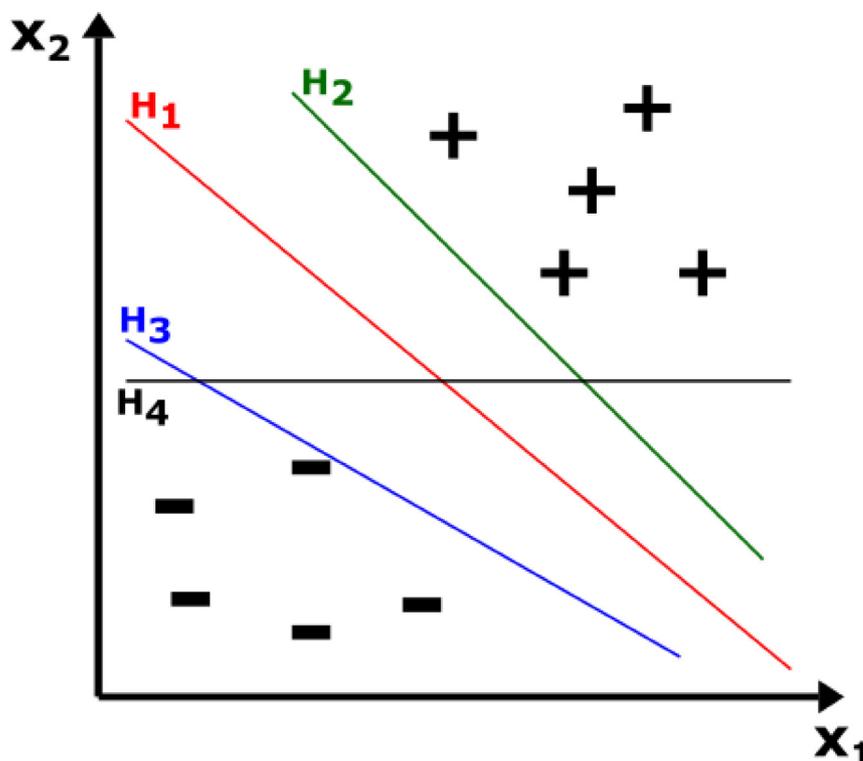
O SVM é responsável por encontrar a melhor fronteira de separação entre classes e rótulos possíveis para um dado conjunto de dados que sejam linearmente separáveis (REMIGIO, 2020). O SVM baseia-se na ideia de encontrar um hiperplano que melhor separa os recursos em diferentes domínios (YADAV, 2018). Então, o SVM tem como objetivo encontrar o hiperplano para um determinado conjunto de dados, cujas classes são linearmente separáveis (REMIGIO, 2020). Essa ideia pode ser visualizada na Figura 7.

2.4.2 *Stochastic Gradient Descent*

O gradiente descendente é uma técnica para otimizar funções complexas através de um programa computacional. Seu objetivo é: dada alguma função arbitrária, encontrar um mínimo (GOODFELLOW *et al.*, 2016). Matematicamente, o gradiente descendente pode ter apenas um único mínimo para alguns pequenos subconjuntos de funções, aqueles que são convexos. Para as funções mais realistas, pode haver muitos mínimos, então a maioria dos mínimos são locais (GOODFELLOW *et al.*, 2016).

Quanto maior for o gradiente, mais íngreme será a inclinação (GEEKS, 2021). Essa inclinação pode ser observada no exemplo da Figura 8. O objetivo do algoritmo de gradiente descendente é encontrar o valor de 'x' de tal forma que 'y' é mínimo.

O gradiente descendente estocástico, do inglês *Stochastic Gradient Descent* (SGD), é um

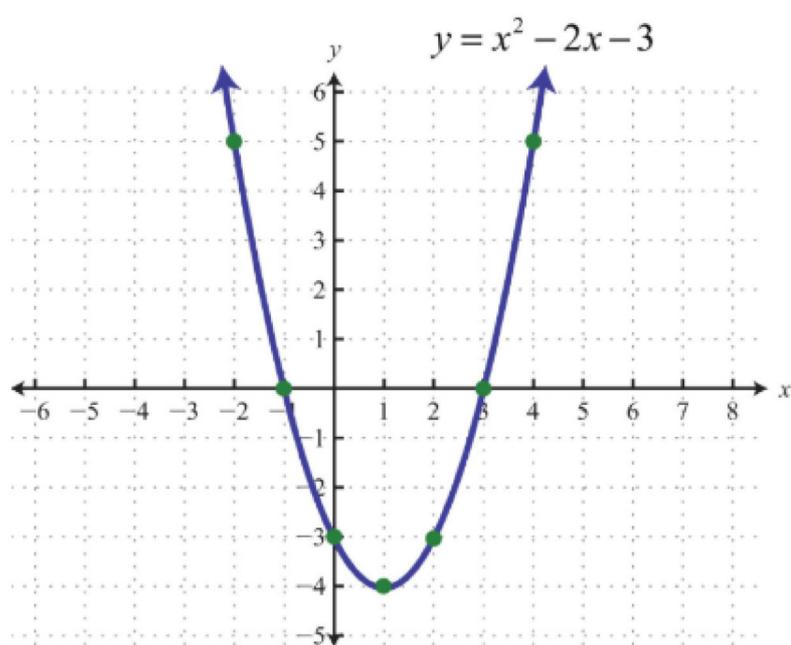
Figura 7 – Possíveis hiperplanos de separação para um *dataset* bidimensional e binário.

Fonte: Remigio (2020)

algoritmo popular utilizado em várias técnicas de aprendizado de máquina (SRINIVASAN, 2019). Por ser um algoritmo estocástico, ele está vinculado a uma probabilidade aleatória (AZEVEDO *et al.*, 2020). O SGD tem sido aplicado com sucesso em problemas de aprendizagem de máquina em larga escala e esparsos. Ele é frequentemente encontrado na classificação de texto e em PLN (SCIKITLEARN, 2022). O SGD calcula o gradiente usando uma única amostra, tornando-o menos complexo e custoso que outros algoritmos.

Como uma iteração do algoritmo gradiente descendente requer uma previsão para cada instância no conjunto de dados de treinamento, ele pode demorar muito quando se tem muitos milhões de instâncias. Visto isso, o SGD é uma variação onde a atualização para os coeficientes é realizada para a instância de treinamento, em vez do final do lote de instâncias (GOODFELLOW *et al.*, 2016).

Figura 8 – Uma função parabólica com duas dimensões.



A parabolic function with two dimensions (x,y)

Fonte: Srinivasan (2019)

3 METODOLOGIA

O principal diferencial deste trabalho é o analisador e corretor verbal utilizando métodos tradicionais e intuitivos de PLN. Neste capítulo, são descritos os materiais e a metodologia utilizados para a realização das tarefas propostas.

3.1 Materiais

Nesta seção, são descritas as ferramentas adotadas neste trabalho. Essas ferramentas foram usadas para a aquisição de base de dados e também para a manipulação de tarefas de PLN. São descritas as bases de dados e as bibliotecas usadas para o desenvolvimento do trabalho.

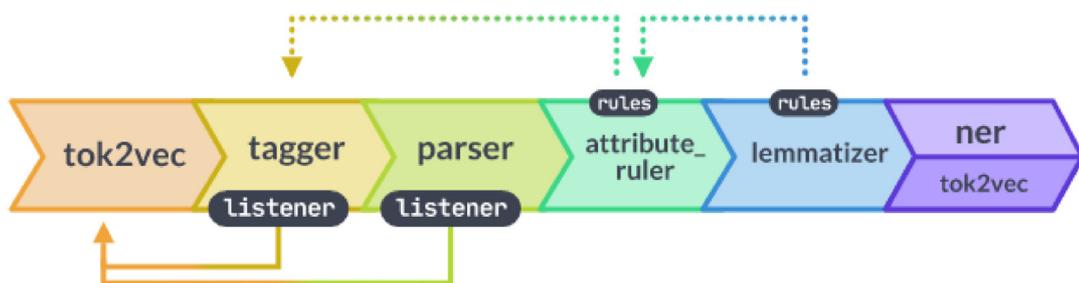
Biblioteca SpaCy

O SpaCy é uma biblioteca para processamento avançado de linguagem natural em Python e Cython. Foi construído com base nas pesquisas mais recentes e foi projetado desde o primeiro dia para ser usado em produtos reais (HONNIBAL *et al.*, 2020). Ela apresenta modelos de redes para diversas tarefas de PLN, possuindo *pipelines* pré-treinadas e oferecendo suporte de *tokenização* e treinamento para mais de 60 idiomas. Essa ferramenta dispõe de meios para realizar e auxiliar na análise léxica, morfológica e sintática para frases em português e em outros idiomas.

O SpaCy contém três *pipelines* treinados em português, que, assim como as das outras linguagens, são projetados para serem eficientes em termos de velocidade e tamanho, além de seus componentes dependerem uns dos outros, que é algo que tem que ser avaliado se for necessário realizar alguma adaptação ou alteração (EXPLOSION, 2021). As Figuras 9 e 10 mostram o *design* de uma das *pipelines* do SpaCy, no qual a primeira é específica sobre os componentes do SpaCy e a segunda mostra mais claramente um *design* de acordo com as tarefas.

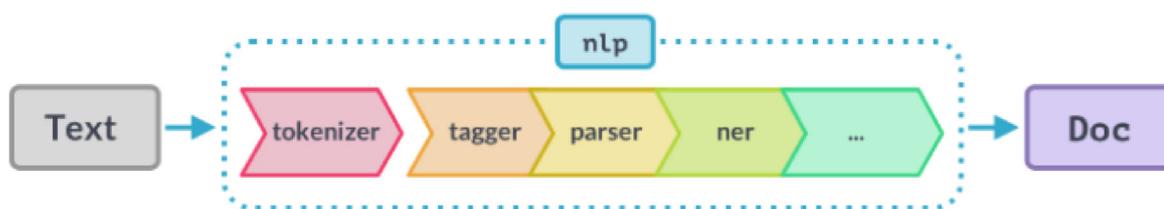
A *pipeline* do SpaCy utilizada neste trabalho foi criada a partir do banco de árvores criado por Rademaker *et al.* (2017), que é baseado na versão convertida da gramática de restrições do

Figura 9 – Design do Pipeline do SpaCy.



Fonte: Explosion (2021)

Figura 10 – Exemplo de Design do Pipeline de PLN.



Fonte: Abhisheek (2020)

bosque, que faz parte do banco de dados Floresta Sintá(c)tica. Através dessa *pipeline*, o sistema é capaz, por exemplo, de *tokenizar* (*tok2vec*), classificar os *tokens* segundo a sua classe gramatical (*tagger*) e criar uma árvore sintática (*parser*).

Biblioteca *mlconjug3*

O *mlconjug3* (DIAO, 2021) é uma biblioteca da linguagem Python para conjugação de verbos em português e em outras línguas usando técnicas de aprendizado de máquina. Essa biblioteca é capaz de gerar verbos conjugados em todos os tempos verbais. Nela, é disponibilizado um modelo pré-treinado de conjugação para a língua portuguesa.

A Tabela 3 contém um exemplo de conjugação gerado pela biblioteca *mlconjug3*. Nessa tabela, o verbo 'mostrar' é apresentado na primeira pessoa do plural e conjugado nos modos: indicativo, condicional, conjuntivo, infinitivo e imperativo.

Essa biblioteca contém uma *application programming interfaces* (API) que possibilita interagirmos como os modelos de aprendizado de máquina dela (DIAO, 2021). Através dessa API, é possível a criação de um novo modelo de conjugação verbal. Uma das funções, por exemplo, é um extrator personalizado que extrai recursos dos verbos. Além dela, outra função importante é classe "*Conjugator*", que gerencia o conjunto de dados Verbiste (SARRAZIN, 2020) e fornece uma interface com o pipeline *scikit-learn*, essa classe define o método conjugado (verbo e linguagem) que é o principal método do módulo (DIAO, 2021).

A "*Conjugator*" é a classe principal do projeto *mlconjug3*. A classe gerencia o conjunto de dados Verbiste e fornece uma interface com o pipeline *scikit-learn*. Se não forem fornecidos parâmetros, a linguagem padrão será definida como francês e o pipeline de conjugação francês pré-treinado é usado. A classe define o método conjugado (verbo, linguagem) que é o principal método do módulo.

Biblioteca *Natural Language Toolkit*

O *Natural Language Toolkit* (NLTK), segundo Loper e Bird (2002), é um conjunto de programas, módulos e tutoriais de código aberto, fornecendo dados computacionais prontos para uso material didático de linguística. O NLTK também contém os corpus utilizados por este sistema para auxiliar na correção de erros verbais, um deles é chamado de Mac-Morpho e o

Tabela 3 – Exemplo de Conjugação do *mlconjug3*

Modo	Tempo	Verbo
Indicativo	Indicativo presente	mostramos
Indicativo	Indicativo Pretérito Perfeito Composto	mostrado
Indicativo	Indicativo pretérito imperfeito	mostrávamos
Indicativo	Indicativo Pretérito Mais que Perfeito Composto	mostrado
Indicativo	Indicativo Pretérito Mais que Perfeito Simples	mostráramos
Indicativo	Indicativo pretérito perfeito simples	mostramos
Indicativo	Indicativo Pretérito Mais que Perfeito Anterior	mostrado
Indicativo	Indicativo Futuro do Presente Simples	mostraremos
Indicativo	Indicativo Futuro do Presente Composto	mostrado
Condicional	Condicional Futuro do Pretérito Simples	mostraríamos
Condicional	Condicional Futuro do Pretérito Composto	mostrado
Conjuntivo	Conjuntivo Subjuntivo Presente	mostremos
Conjuntivo	Conjuntivo Subjuntivo Pretérito Perfeito	mostrado
Conjuntivo	Conjuntivo Subjuntivo Pretérito Imperfeito	mostrássemos
Conjuntivo	Conjuntivo Subjuntivo Pretérito Mais que Perfeito	mostrado
Conjuntivo	Conjuntivo Subjuntivo Futuro Simples	mostrarmos
Conjuntivo	Conjuntivo Subjuntivo Futuro Composto	mostrado
Infinitivo	Infinitivo Pessoal Presente	mostrarmos
Infinitivo	Infinitivo Pessoal Pretérito	mostrado
Imperativo	Imperativo Afirmativo	mostremos
Imperativo	Imperativo Negativo	não mostremos

Fonte: Do autor

outro é o Floresta Sintá(c)tica.

O Mac-Morpho, como salienta Fonseca e Rosa (2013), é composto por textos em português do jornal Folha de São Paulo, que, por se tratar de artigos jornalísticos, o corpus contém orações sobre diversos assuntos, como esportes e política. Já o Floresta Sintá(c)tica é um corpus anotado que contém textos do português do Brasil e de Portugal, pertencente a um projeto entre

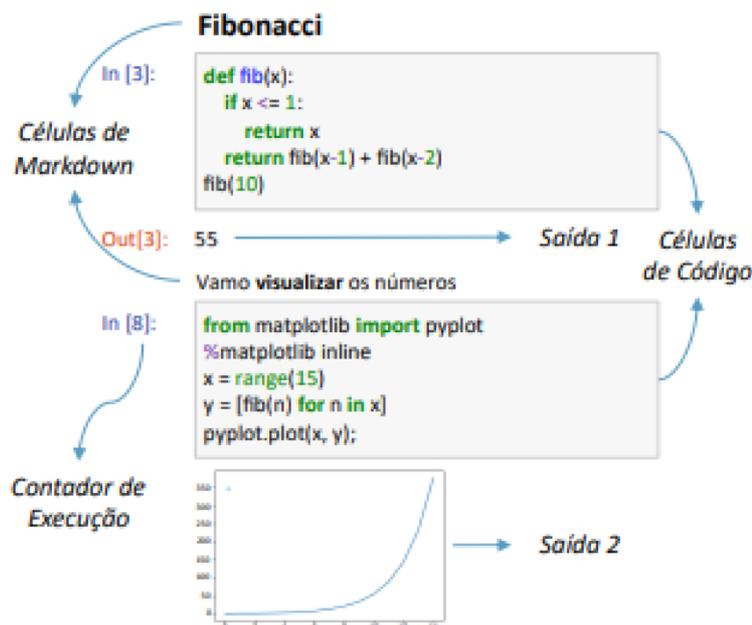
a Linguateca e o projeto VISL (FILHO, 2015).

Sistema Jupyter Notebook

Por este trabalho necessitar de um tratamento de dados, ele foi feito através da ferramenta Jupyter Notebook. Segundo Pimentel *et al.* (2021), o Jupyter Notebook é um sistema que pode ser usado de forma que os resultados vão sendo exibidos de forma instantânea e sendo documentado durante o seu desenvolvimento. Por conta disso, ele é ideal para acompanhar e visualizar os diversos estágios de trabalhos como este.

O Jupyter Notebook pode produzir os seguintes tipos de documentos em suas células: código executável, texto formatado (*markdown*), *raw NBconvert* e texto simples. Os resultados de uma execução de uma célula de código são armazenados e exibidos interativamente durante a execução do notebook (PIMENTEL *et al.*, 2021). Por conta da interatividade do Jupyter, suas células não necessitam ser executadas em uma ordem específica, como ocorre normalmente em arquivos de código. A Figura 11 contém exemplos de células do Jupyter Notebook.

Figura 11 – Exemplo de notebook executado com *markdown*, código e resultados.



Fonte: Pimentel *et al.* (2021)

A primeira célula exposta na figura é uma *markdown*, que conta com a palavra 'Fibonacci' destacada em negrito. Já a segunda célula, contém uma função em Python para o cálculo da sequência de fibonacci, e a chamada dessa função. Logo em seguida há uma saída, sendo resultado do décimo elemento da sequência de fibonacci, o número 55. Após isto, aparece uma célula com um texto simples. A última célula é um código executável para a criação de um gráfico da sequência de fibonacci e sua saída sendo o referido gráfico.

Outro aspecto a ser levado em consideração nessa figura são os contadores de execução. Cada execução no Jupyter Notebook é contabilizado. Visto que as células não necessitam serem executadas em uma ordem específica, essa numeração auxilia o usuário a visualizar a ordem das células já executadas.

3.2 Métodos

Nesta seção, são descritos os métodos adotados para o desenvolvimento deste trabalho. Primeiramente, a sentença recebida pelo sistema passa por uma análise gramatical e tem seu verbo e sujeito reconhecidos. Após isto, o sistema realiza uma verificação de erros gramaticais, e, caso tenha a ocorrência de algum erro, efetua a seleção do verbo mais provável entre candidatos gerados automaticamente. Na Figura 12, é possível observar o fluxograma do corretor verbal.

Este trabalho também pode ser entendido como dividido em duas partes: análise de erros verbais e correção de erros verbais. A visão bipartida foi adotada para esta explanação, visto que, dependendo do resultado da primeira parte, ou seja, se a sentença contém erro de concordância verbal ou não, a segunda parte do trabalho nem é acionada pelo sistema. A seguir são detalhadas essas partes.

3.2.1 Análise de erros verbais

Após o recebimento de uma oração, o sistema a submete à biblioteca SpaCy. Por meio dessa biblioteca, o processo da análise léxica é iniciado, fazendo a *tokenização* da oração, ou seja, transformando cada palavra e pontuação em *tokens*. Facilitando assim, a interpretação do sistema, que em seguida realiza a análise morfológica da sentença, reconhecendo a classe gramatical de cada *token*. Nas Tabelas 4 e 5, isso é demonstrado.

Tabela 4 – Exemplo de *tokenização*

Oração	Oração Processada em Tokens
Maria e João são irmãos	'Maria', 'e', 'João', 'são', 'irmãos'

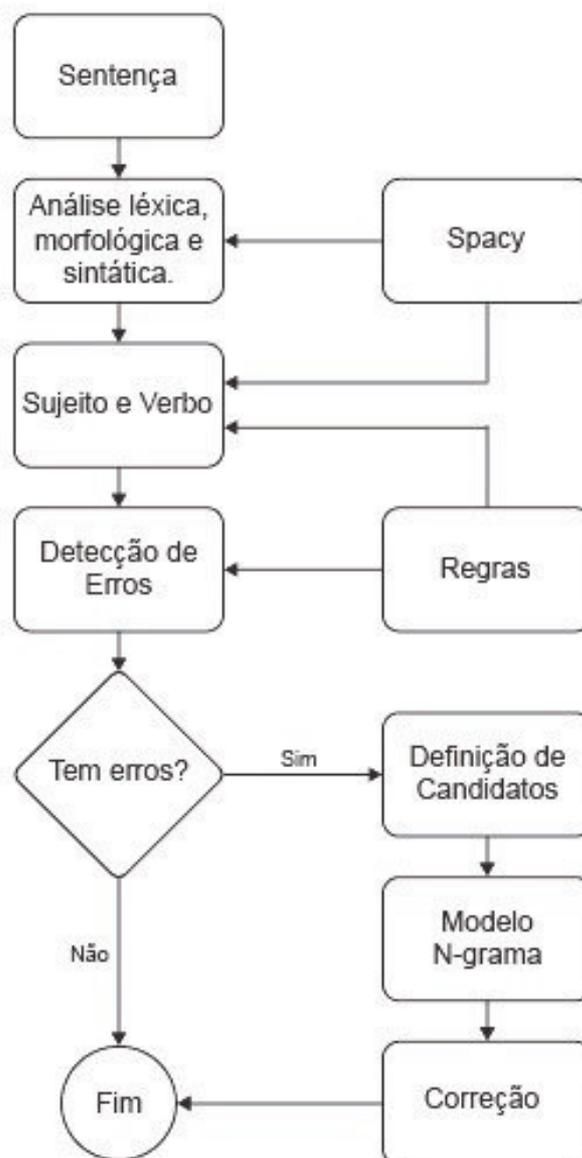
Fonte: Adaptado de Sakurai (2019)

Tabela 5 – Exemplos de POS tagging

Token	Classe Gramatical
Maria	PROPN
e	CCONJ
João	PROPN
são	AUX
irmãos	NOUN

Fonte: Do autor

Figura 12 – Fluxograma do analisador e corretor verbal.

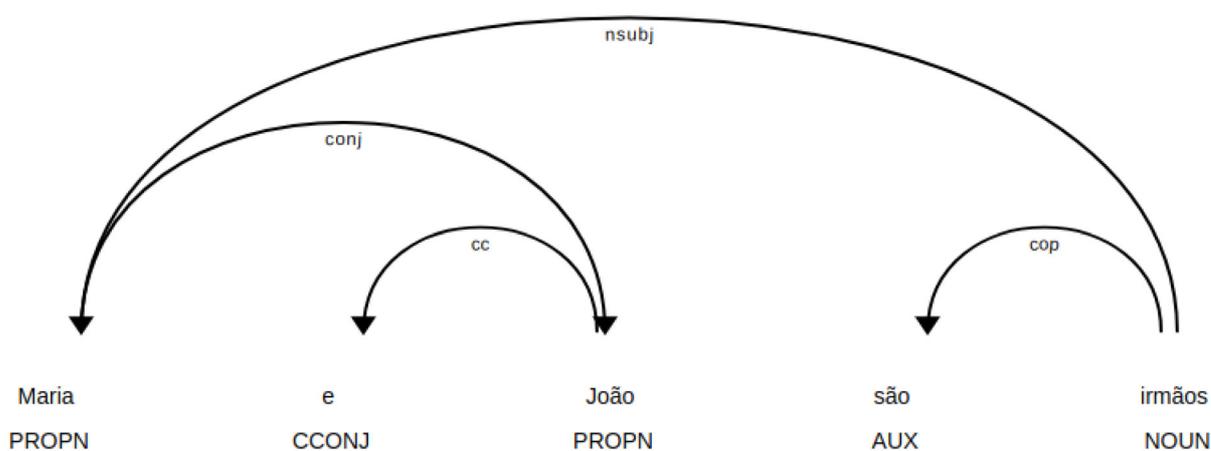


Fonte: Do autor

Após as fases de análise léxica e morfológica, o *parsing* pode ser realizado. Nesse ponto, a árvore sintática é criada, e que, a ligação entre as palavras é entendida pelo sistema. Sendo assim, o sujeito, verbo principal, predicado e o objeto da sentença, por exemplo, podem ser reconhecidos, tornando possível a análise verbal da oração.

Para a análise verbal da sentença, foi utilizada uma abordagem baseada em regras, especificamente o SVC. Portanto, a partir da palavra rotulada como 'nsubj' na árvore de dependência, é reconhecido o núcleo do sujeito, e, se busca em suas ligações, outros substantivos e pronomes, para o caso do sujeito ser composto. Na Figura 13, é possível observar que o núcleo do sujeito da oração 'Maria e João são irmãos' é 'Maria', e tem uma ligação rotulada como 'conj' com 'João', indicando que o sujeito tem mais de um elemento.

Figura 13 – Exemplo de árvore sintática com sujeito composto.



Fonte: Do autor

Neste ponto, o SVC deste sistema busca primeiro reconhecer se o sujeito é simples ou composto. O reconhecimento do tipo de sujeito da oração é essencial para a definição das regras que serão usadas para a avaliação da sentença, pois as regras para tais tarefas diferem.

Após ser reconhecido se o sujeito é simples ou composto, o sistema compara se o verbo concorda com o sujeito em pessoa e número, pois a concordância verbal é baseada na relação entre o sujeito e verbo, onde o segundo precisa concordar com o sujeito em número e pessoa. Inclusive, a análise do SVC se estende aos outros verbos ligados ao verbo que está conectado ao sujeito. Esse reconhecimento ocorre a partir da análise morfológica de cada *token*, realizada pela biblioteca SpaCy. Através disso, o sistema é capaz de reter informações, como se uma determinada palavra ou verbo está no singular ou no plural. Na Tabela 6, a morfologia gerada pelo SpaCy pode ser observada.

Tabela 6 – Exemplos da análise morfológica do SpaCy

Token	Gênero	Número	Pessoa	Tempo
Maria	Feminino	Singular	-	-
e	-	-	-	-
João	Masculino	Singular	-	-
são	-	Plural	3	Presente
irmãos	Masculino	Plural	-	-

Fonte: Do autor

No caso da oração 'Maria e João são irmãos', o sujeito aponta para um verbo na terceira pessoa do plural, indicando que a oração está correta. O sujeito, que é 'Maria e João', contém dois núcleos que são substantivos próprios e estão no singular, mas que, por estarem juntos no sujeito da oração, exigem um verbo no plural e na terceira pessoa. Como este trabalho adota a regra geral da concordância verbal, toda vez que o sujeito for composto, ele exigirá que o verbo esteja no plural.

3.2.2 Corretor de Erros Verbais

Assim que um erro é reconhecido pelo sistema, a correção de erros é acionada. Então, ocorre a *lematização* do verbo que foi reconhecido como conjugado errado. 'A *lematização* é o processo de expor determinada palavra por meio do infinitivo dos verbos e masculino singular dos substantivos e adjetivos (LUCCA; NUNES, 2002). Na Tabela 7, estão alguns exemplos de verbos *lematizados*.

Tabela 7 – Exemplos de *lematização*

Verbo	Verbo <i>lematizado</i>
Vivendo	Viver
Tentando	Tentar
Falei	Falar
Apresentei	Apresentar
Estudava	Estudar

Fonte: Do autor

Após o processo de *lematização* do *token*, usa-se a biblioteca *mlconjug3* para gerar uma série de candidatos a partir da conjugação desses verbos. Nesse ponto, foi criado um novo modelo de conjugação verbal a partir da API do *mlconjug3*. Nesse novo modelo, a base de dados Verbiste para o português foi utilizada.

Para a criação de um novo modelo, a *pipeline* da função *mlconjug3.Model* necessita de um vetorizador de recursos, de um seletor de recursos e de um classificador (DIAO, 2021). O vetorizador de recursos utilizou a própria função de extração da biblioteca. Os algoritmos lineares LinearSVC e SGDClassifier foram usados para a seleção de recursos e classificação, respectivamente. O LinearSVC utilizou a penalidade 'l2' e o SGDClassifier, a penalidade 'ElasticNet' (SCIKITLEARN, 2021).

Nesse modelo, a base de dados foi dividida em 80% para treinamento e 20% para teste, como usual na literatura (LEAL *et al.*, 2019). A janela deslizante de faixa de n-grama para o vetorizador foi de (2,7). Para o algoritmo SVC e para o SDGClassifier, respectivamente, os seguintes parâmetros comuns foram ajustados:

SVC penalty = "l2", max_iter=12000.

SGD penalty = 'elasticnet', l_ratio = 0.15, max_iter = 4000, alpha = 1e-5, loss = "log".

Nesse conjugador, a pessoa e o número são escolhidos de acordo com o que é exigido pelo sujeito da oração que o sistema está analisando. Por exemplo, na oração 'Os jornalistas mostra os dados', existe um erro de concordância, pois o sujeito, 'jornalistas', requer um verbo na terceira pessoa do plural, portanto, o sistema gera os seguintes candidatos:

- mostram;
- mostravam;
- mostrados;
- mostraram;
- mostrarão;
- mostrariam;
- mostrem;
- mostrassem;
- mostrarem.

No caso do verbo acima, existem palavras que podem ser vistas nos diferentes modos e tempos verbais, como é o caso de 'mostrem', que é tanto subjuntivo presente, quanto o imperativo afirmativo. Em casos como esse, o sistema conta o *token* em específico, não o modo e tempo verbal. Esse modelo obteve 98% de precisão, que é um desempenho já comum para a API do *mlconjug3* (DIAO, 2021).

Nascimento *et al.* (2021) explicam que o tempo mais que perfeito simples, ainda que presente nos livros didáticos, está em estágio de mudança linguística, visto que a sua variante composta sobrepõe-se a ela, deixando-a em desuso. Portanto, neste trabalho, optou-se por retirar o modo mais que perfeito simples dos candidatos a correção, principalmente, por se tratar de uma correção probabilística.

O tempo verbal imperativo negativo, assim como o mais que perfeito, não é enquadrado entre os candidatos a correção neste sistema. Pois, no imperativo negativo, o verbo vem precedido pela palavra 'não', como neste trabalho foi desenvolvido a nível de *tokens*, essa negativa será observada dentro da oração analisada, auxiliando a escolha ou não de um determinado candidato.

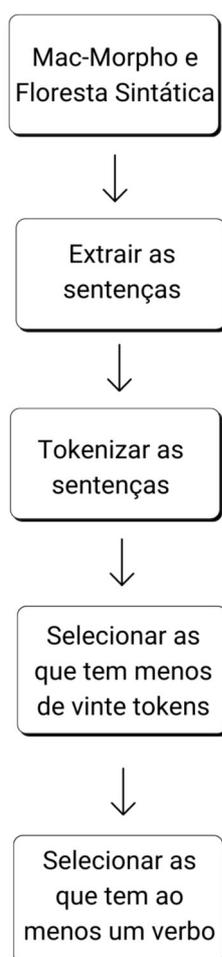
Assim que os candidatos são selecionados, o sistema passa para a fase da seleção de qual candidato é o mais adequado e provável para corrigir o erro de concordância verbal. Para tanto, o método estatístico n-grama é o escolhido.

O modelo n-grama, assim como qualquer modelo estatístico, tem um desempenho afetado de acordo com a base de dados, tanto em seu tamanho (quantidade de *tokens*), como em sua variedade (variedades de *tokens*). Não obstante, neste sistema, o n-grama foi auxiliado pela série de candidatos gerados pelo *mlconjug3*, contribuindo assim para uma utilização mais direcionada do modelo.

As base de dados Mac-Morpho e a Floresta Sintá(C)tica, presentes no NLTK, foram usadas para a criação de um modelo probabilístico n-grama. A base Mac-Morpho, que segundo Fonseca *et al.* (2015), contém em torno de um milhão de *tokens*, não foi usada por completo, por não ser necessário para o escopo deste projeto. De igual modo, também só uma parte do Floresta Sintá(C)tica foi utilizada.

A fim de evitar o problema de uma grande quantidade de dados irrelevantes, foi realizada uma seleção na base de dados Mac-Morpho e do Floresta Sintá(c)tica. Nessa seleção, optou-se por selecionar somente sentenças que têm pelo menos um verbo. Para a extração das sentenças úteis nas bases de dados, uma tarefa baseada em regras foi utilizada. O primeiro passo foi receber as sentenças, depois *tokenizar* cada sentença, após isso, selecionar somente as sentenças com no máximo vinte *tokens* e que contam com pelo menos um verbo. Esse processo pode ser visualizado na Figura 14.

Figura 14 – Extração das sentenças das bases de dados do NLTK.



Fonte: Do autor

Para uma melhor avaliação do método, duas formas de extração de n-gramas foram utilizadas. Na primeira, os n-gramas foram criados agrupando os *tokens* vizinhos a POS tags marcadas como verbos, o qual continuará a ser chamado neste trabalho somente como modelo neighbor-grama. Já na segunda, os n-gramas foram extraídos agrupando o verbos às classes gramaticais do seu sujeito, o qual será chamado no restante do decorrer deste trabalho como modelo skip-grama.

Além desses dois modelos que foram aproveitados para a seleção do candidato mais provável, criou-se outro modelo para auxiliar nos casos em que existe um advérbio ligado ao verbo analisado. Portanto, ligou-se o advérbio ao tempo verbal do n-grama, assim, diminuindo o número de candidatos a serem avaliados. Esse processo segue uma extração semelhante ao do skip-grama, pois ela também utiliza a árvore de dependência, conforme observado a seguir.

Um exemplo da diferença entre as formas como os n-gramas foram extraídos neste trabalho, pode ser observado a partir da seguinte sentença: "João e Maria dormiram enquanto eu discursava no evento". Nesse caso, os n-gramas foram extraídos a partir dos verbos: "dormiram" e "discursava". No modelo neighbor-grama, os verbos e seus vizinhos foram utilizados para a sua criação, como apontado na Figura 15. Já no modelo skip-grama, foram aproveitados e os verbos e as classes gramaticais do sujeito, como demonstrado na Figura 16.

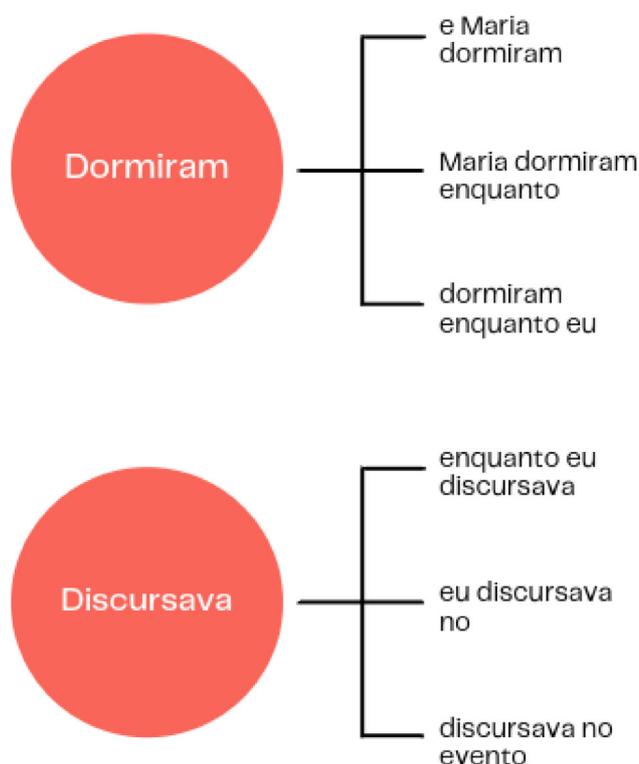
Os modelos n-grama tem como objetivo calcular a probabilidade de palavras aparecerem juntas em uma sentença. Portanto, no modelo neighbor-grama foram contabilizados de uma forma que eles são contados tanto conjunta, quanto individualmente. Por exemplo, no modelo neighbor-grama 'vai até domingo', tanto o bi-grama 'até domingo', quanto os uni-gramas 'até' e 'domingo', vão ter suas aparições contabilizadas ao lado do verbo 'vai'. Já no modelo skip-grama, isso não ocorre, contabilizando somente um n-grama por verbo.

Na linguagem Python, existe uma estrutura de dados chamada de dicionário, que é uma coleção de dados não ordenados, que ao contrário de outros tipos de dados que contém somente um valor, armazena um par de valores-chave (W3SCHOOLS, 2022). Já o *defaultdict*, que é uma subclasse do dicionário, armazena um objeto semelhante a um dicionário (GEEKS, 2022). Essa subclasse foi manuseada para a criação dos modelos de n-grama deste sistema.

Na Tabela 8, demonstra-se como os bi-gramas e uni-gramas são contabilizados no modelo de neighbor-grama. Pois, como pode ser observado, primeiro eles são contados com o número de ocorrências para cada tri-grama. Nesse caso, na oração 'eu estou feliz pois eu estou vencendo', o bi-grama 'eu estou' tem uma ocorrência tanto com o uni-grama 'feliz', quanto com o 'vencendo'. Algo semelhante ocorre no modelo skip-grama

A Tabela 9 mostra os tri-gramas da oração 'eu estou feliz porque eu estou vencendo', de forma probabilística. Assim, devido ao fato dos uni-gramas 'feliz' e 'vencendo' formarem um tri-grama com o mesmo bi-grama, 'eu estou', ambos terão 50% de probabilidade de aparecerem juntos desse bi-grama.

Figura 15 – Exemplo de extração do modelo tri-grama.



Fonte: Do autor

Figura 16 – Exemplo de extração do modelo skip-grama.



Fonte: Do autor

Dentro do sistema, o modelo neighbor-grama é aproveitado de forma que o verbo em específico que está sendo analisado como candidato tenha sua probabilidade calculada com diferentes n-gramas (tri-gramas e bi-gramas). Na Figura 17, há um exemplo da saída gerada pelo modelo de n-grama feito na linguagem Python. É mostrada no exemplo a probabilidade de cada palavra estar relacionada a palavra 'evento'.

A Figura 18 contém um exemplo de pesquisa de probabilidade de um bi-grama no modelo neighbor-grama, no caso, apresenta-se uma pesquisa do par 'foi por'. Como é possível observar, esse par tem um significativo número de ocorrências, por isso, na captura de tela, não é possível visualizar todas as palavras

Tabela 8 – Tabela com a contagem de tri-gramas

	estou	feliz	pois	eu	vencendo	.
('eu', 'estou')	0	1	0	0	1	0
('estou', 'feliz')	0	0	1	0	0	0
('feliz', 'porque')	0	0	0	1	0	0
('porque', 'eu')	1	0	0	0	0	0
('estou', 'vencendo')	0	0	0	0	0	1

Fonte: Do autor

Tabela 9 – Tabela de probabilidade dos tri-gramas

	estou	feliz	pois	eu	vencendo	.
('eu', 'estou')	0	0.5	0	0	0.5	0
('estou', 'feliz')	0	0	1	0	0	0
('feliz', 'pois')	0	0	0	1	0	0
('porque', 'eu')	1	0	0	0	0	0
('estou', 'vencendo')	0	0	0	0	0	1

Fonte: Do autor

Figura 17 – Screenshot da saída da pesquisa no modelo n-grama.

```
1 dict(model_tri[('evento')])
{'começa': 0.09523809523809523,
 'mostrou': 0.047619047619047616,
 'tem': 0.047619047619047616,
 'será': 0.14285714285714285,
 'irá': 0.047619047619047616,
 'faz': 0.047619047619047616,
 'termina': 0.047619047619047616,
 'é': 0.047619047619047616,
 'ofereceu': 0.047619047619047616,
 'conta': 0.047619047619047616,
 'contará': 0.047619047619047616,
 'acontece': 0.09523809523809523,
 'oferecerá': 0.047619047619047616,
 'reúne': 0.047619047619047616,
 'inclui': 0.047619047619047616,
 'foi': 0.047619047619047616,
 'terá': 0.047619047619047616}
```

Fonte: Do autor

As consultas de probabilidade do modelo skip-grama não são diferentes do modelo neighbor-grama. Na Figura 19, é possível observar duas células e suas saídas. A primeira contém a pesquisa da probabilidade de duas palavras rótuladas como 'NOUN' formarem o sujeito do verbo 'querem'. Já na segunda célula, é apresentada a probabilidade desse verbo e um par de POS tags estarem conectados.

Cada verbo é testado com no máximo três combinações de bi-gramas, que representam os

Figura 18 – *Screenshot* da saída da pesquisa de um bi-grama no modelo n-grama.

```
1 dict(model_tri[('foi', 'por')])
'julgado': 0.005988023952095809,
'baixada': 0.005988023952095809,
'atingida': 0.005988023952095809,
'preso': 0.011976047904191617,
'proposta': 0.011976047904191617,
'atingido': 0.023952095808383235,
'descrita': 0.005988023952095809,
'surpreendido': 0.005988023952095809,
'provocado': 0.005988023952095809,
'condenado': 0.005988023952095809,
'carregada': 0.005988023952095809,
'adiada': 0.005988023952095809,
'dominado': 0.011976047904191617,
'morto': 0.011976047904191617,
'facilitada': 0.005988023952095809,
'utilizado': 0.005988023952095809,
'identificado': 0.005988023952095809,
'desenvolvido': 0.005988023952095809,
'cedido': 0.005988023952095809,
```

Fonte: Do autor

Figura 19 – *Screenshot* da saída da pesquisa no modelo skip-grama.

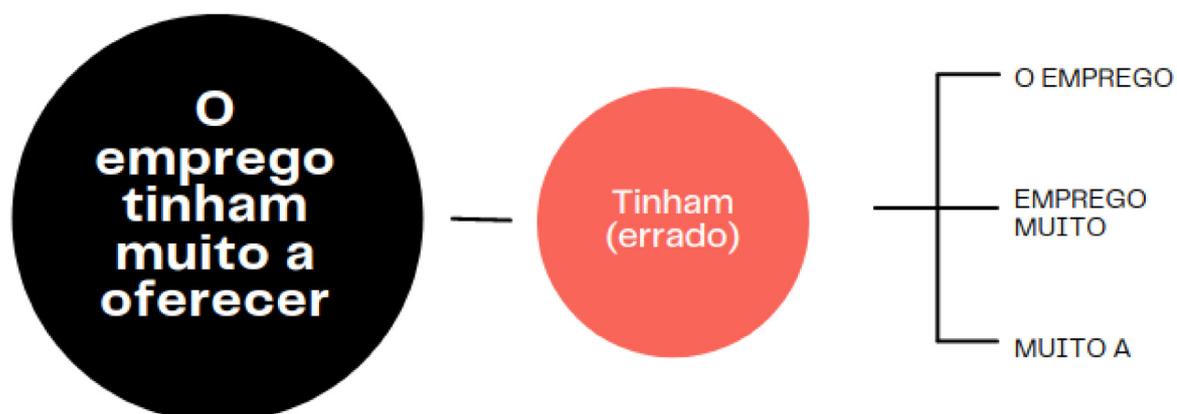
```
In [18]: 1 dict(model_ngram[('NOUN'), ('NOUN')])[('querem')]
Out[18]: 0.0007468259895444362

In [19]: 1 dict(model_ngram[('NOUN'), ('PRON')])[('querem')]
Out[19]: 0.0013458950201884253
```

Fonte: Do autor

vizinhos do verbo errado. Por exemplo, na oração 'O emprego tinham muito a oferecer', existe um erro de concordância, então os candidatos à correção seriam testados junto aos seguintes bi-gramas expostos na Figura 20. Depois da escolha do candidato mais provável, o sistema realiza a troca, inserindo-o na sentença.

Figura 20 – Bi-gramas vizinhos ao verbo.



Fonte: Do autor

4 RESULTADOS

Este capítulo trata da apresentação dos resultados obtidos das diferentes tarefas realizadas durante este trabalho. Os testes descritos neste capítulo foram utilizados para a avaliação e a análise da metodologia. Este estudo tem duas tarefas principais como objetivo, a detecção e a correção de erros verbais, portanto foram usadas métricas comuns em outros trabalhos que envolvem n-grama e abordagens baseadas em regras.

Para a avaliação das tarefas, foi utilizada uma base que se encontra no *site* Kaggle (CORREA, 2022) e também uma base criada manualmente. A base de frases do Kaggle contém 347 orações, sendo 36% delas com orações contendo erros de concordância verbal. A base de testes criada para este trabalho contém 1879 orações, contendo 1267 orações erradas e 612 orações corretas. Totalizando uma base de teste com 2.226 orações.

Na base criada manualmente para os testes desta pesquisa, foram utilizados sete dos nomes próprios mais comuns registrados no Brasil em 2021 para a elaboração dos sujeitos (TADEU, 2021). Esses nomes são os seguintes:

- Miguel;
- Arthur;
- Helena;
- Alice;
- Laura;
- Davi.

Também destaca-se que foram extraídas do Mac-Morpho e do Floresta alguns sujeitos, verbos e objetos de forma aleatória para somar a base de teste.

Depois de selecionar os sujeitos, os verbos e os objetos, foram geradas automaticamente diversas orações. Sendo que para cada sentença com um erro de concordância verbal, pelo menos um exemplo de correção também deveria ser gerado. Por fim, foi realizada uma revisão manual de cada uma das orações.

Para o conjunto de orações da base de teste, uma nuvem de palavras foi gerada para a visualização da diversificação de palavras presentes na base. Uma nuvem de palavras é um recurso visual que apresenta as palavras de um conjunto de dados de forma que o tamanho das palavras indica sua frequência (VASCONCELLOS-SILVA; ARAUJO-JORGE, 2019). Na Figura

Para a avaliação de detecção de erros verbais, este trabalho utilizou métricas comuns na análise de linguagens com poucos recursos de PLN (WIECHETEK *et al.*, 2021). Portanto, as métricas escolhidas para a avaliação são: precisão, *recall* e *f-score* (F1). Essas métricas são demonstradas nas equações abaixo (4.1, 4.2 e 4.3). Nessas equações, VP são os verdadeiros positivos, FP, os falsos positivos e FN, os falsos negativos.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (4.1)$$

$$\text{Recall} = \frac{VP}{VP + FN} \quad (4.2)$$

$$F_1 = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (4.3)$$

Importante frisar que os verdadeiros positivos, nesse caso, são aqueles marcados como errados e que o sistema teve sucesso em apontar como tal. Falsos positivos são os marcados como corretos mas que não foram classificados dessa forma pelo sistema. Por fim, os falsos negativos são orações erradas que a ferramenta falhou em reconhecer como erradas.

A precisão é uma métrica que busca representar qual a proporção de identificações positivas foi realmente correta (FERREIRA, 2019). O *recall*, por sua vez, representa quantos positivos reais foram reconhecidos (PINHEIRO; AMARIS, 2021). Por fim, o *f-score* é uma média harmônica entre precisão e *recall* (VRECH *et al.*, 2021).

Para a avaliação da tarefa de detecção de erros verbais, esta pesquisa foi comparada com trabalhos de verificação de erros presentes na literatura e que contaram com poucos recursos (BHIRUD *et al.*, 2017). Na Tabela 10, o desempenho deste trabalho, nomeado como 'PT', é exposto com outros detectores baseados em regras, feitos para o islandês e para o etíope (BHIRUD *et al.*, 2017). Além deles também, estão presentes na tabela os resultados de um detector criado a partir de redes neurais para o finlandês (WIECHETEK *et al.*, 2021).

Tabela 10 – Resultados da análise de concordância verbal

Métrica	PT	Islandês	Etíope	Finlandês
Recall	94(%)	71.64(%)	94.23(%)	98(%)
Precisão	89(%)	84.21(%)	92.45(%)	79.4(%)
F-score	91(%)	77.41(%)	93.33(%)	87.7(%)

A base de dados de teste contém 1300 orações erradas e o algoritmo teve sucesso em reconhecer 1212. Além disso, a precisão e o *recall* tiveram desempenhos semelhantes ou melhores que em comparação aos outros detectores como para o islandês, finlandês, do etíope.

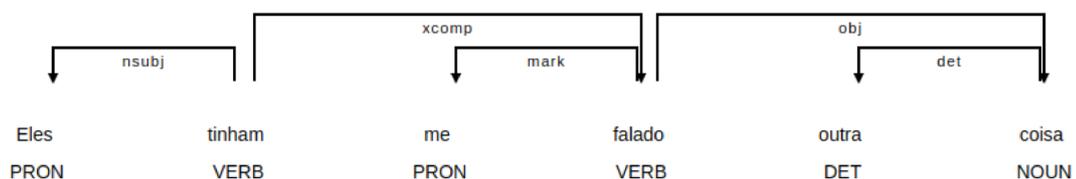
Em razão de poucos exemplos de trabalhos de detecção de erros gramaticais para línguas latinas, resolveu-se comparar este trabalho com outros que também tiveram poucos recursos em seu desenvolvimento. Porém entende-se que esses resultados devem ser considerados com ressalvas, já que não se tratam de línguas da mesma origem que o português. Entretanto, servem para parametrizar os desempenhos dessa tarefa.

Esta tarefa também foi avaliada com erros extraídos da primeira versão escrita deste trabalho. Para tanto, foram selecionados dez parágrafos, com cada um contendo um erro de concordância verbal assinalado pelo orientador desta pesquisa. Ao receber os parágrafos, o detector separou os parágrafos em frases. Desses dez erros assinalados, o detector teve sucesso em encontrar sete. Além disso, o analisador não apontou nenhum erro não apontado pelo orientador.

Duas das falhas de reconhecimento dos erros apontados pelo orientador, ocorreram pelo não reconhecimento do *'et al'*, que está presente em algumas citações indiretas como indicação de uma pluralidade de autores. Esse tipo de erro pode ser corrigido utilizando um reconhecimento de entidades nomeadas que contemple textos acadêmicos.

É importante também destacar as dificuldades da abordagem. Entre elas, estão alguns casos que o SpaCy falha em definir corretamente as ligações, como demonstrado nas seguintes orações: *'Eles tinham me falado outra coisa'* e *'Eles tinha me falado outra coisa'*. Na Figura 23, observa-se a correta ligação entre o pronome *'eles'* e o verbo *'tinham'*, sendo o primeiro o núcleo do sujeito e o segundo, um verbo raiz. Já na Figura 24, observa-se o mesmo pronome sendo conectado erroneamente ao verbo *'falado'*, causando assim, também uma falha no detector de erros. Nele, o detector vai classificar o verbo *'falado'* como erro de concordância.

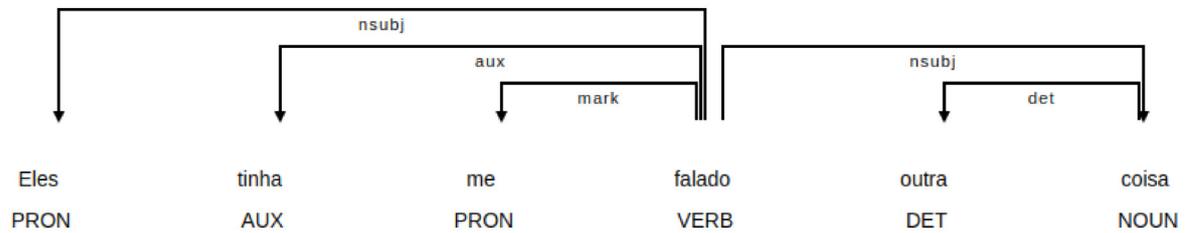
Figura 23 – Árvore de dependência de *'Eles tinham me falado outra coisa'*.



Fonte: Do autor

Uma possível forma de corrigir tal erro seria adicionando novas regras ao detector. Uma das opções seria criar uma nova regra com o auxílio de algumas das classes de extração e reconhecimento de padrões da biblioteca SpaCy, reconhecendo a possibilidade do erro no verbo

Figura 24 – Árvore de dependência de 'Eles tinha me falado outra coisa'.



Fonte: Do autor

mais próximo do sujeito. Outra alternativa é o uso de n-gramas para auxiliar na detecção de erros, reconhecendo combinações anormais (LUNA-RAMÍREZ; JAIMEZ-GONZÁLEZ, 2021).

Entretanto, em razão do desempenho, fica notório que o método baseado em regras foi eficiente na tarefa de reconhecimento dos sujeitos e dos verbos. Demonstrando a que a técnica foi suficiente para o objetivo proposto. Também entende-se que a *pipeline* da biblioteca SpaCy foi essencial para a boa realização da técnica.

Já a correção de erros verbais contou com dois modelos criados a partir do método n-grama. Para diferenciá-los, um foi chamado de neighbor-grama e outro de skip-grama. Os n-gramas do primeiro modelo foram extraídos da relação dos verbos e seus *tokens* vizinhos. Já os n-gramas do segundo foram extraídos dos verbos e as classes gramaticais dos sujeitos ligados a eles.

Na avaliação desses modelos, foram usadas algumas das medidas já observadas na apreciação da detecção de erros, tais como: precisão e *recall*. (BOROŞ *et al.*, 2014). Essas métricas também foram utilizadas em um trabalho de correção de erros sujeito-verbo-concordância para o espanhol, o qual servirá de comparativo para os modelos deste trabalho (BRAVO-CANDEL *et al.*, 2021). Essas duas métricas buscam avaliar o desempenho da correção de erros verbais. Portanto, destaca-se que para isso, a base teve que contar com pelo menos uma alternativa correta de correção. As fórmulas são exemplificadas a seguir.

$$\text{Recall} = \frac{\text{Erros corrigidos}}{\text{Total de erros}} \quad (4.4)$$

$$\text{Precisão} = \frac{\text{Erros corrigidos}}{\text{Erros corrigidos} + \text{Correções erradas}} \quad (4.5)$$

A tarefa de SVC para o espanhol, elaborada por Bravo-Candel *et al.* (2021), também empregou a estratégia de gerar um erro para orações corretas. Portanto, na Tabela 11, podem ser visualizados os desempenhos dos modelos n-gramas desenvolvidos neste trabalho e também os

modelos de representações de palavras (do inglês *word embeddings*) criados a partir de técnicas como GloVe e Word2Vec (BRAVO-CANDEL *et al.*, 2021).

Tabela 11 – Resultados das correções de erros gramaticais

Métrica	Skip-grama	Neighbor-grama	GloVe	Word2Vec
Recall	33(%)	32(%)	19.72(%)	43.01(%)
Precisão	38(%)	35(%)	22.74(%)	55.59(%)

Como pode-se observar, não é uma tarefa com um número tão substancial de acertos quanto a da concordância verbal, em razão da sua complexidade. Também entende-se que muitas vezes existe mais de uma correção correta para uma CEG, principalmente em uma língua rica como o Português (NAPOLES *et al.*, 2015). Dito isso, é possível que a correção realizada, mesmo que certa, não tenha sido contabilizada como correta por não ter um exemplo igual marcado como certo na base de dados. Então, nesse caso, também entende-se que seria oportuno o uso de avaliadores humanos (LIN *et al.*, 2022).

Salienta-se que os modelos comparados aos criados neste trabalho são mais complexos computacionalmente e feitos a partir de uma base de dados maior. A base de dados utilizada pelo trabalho comparativo contém 103.952 sentenças, enquanto os modelos deste trabalho contém 28.134 sentenças. Portanto, é necessário frisar que a faixa de desempenho do trabalho realizado por Bravo-Candel *et al.* (2021) serve como parâmetro para entender que os resultados dos modelos n-gramas foram adequados para a tarefa de CEG.

Apesar de se ter um resultado satisfatório, o n-grama, mesmo que menos custoso que um método de aprendizado de máquina, por ser um método estatístico, tem seu desempenho vinculado ao tamanho da base de dados (PUTRA; ENDA, 2020). Por conta disso, dois fenômenos correlatos devem ser observados: uma limitação de vocabulário, causando casos omissos e também a necessidade de uma base de dados grande.

Porém, entende-se que as desvantagens acima citadas do método n-grama não são de grande empecilho para o escopo do projeto. Pois, não estão entre os objetivos o ensino e o uso de um extenso vocabulário. Contudo, buscava-se o estudo da correção verbal para uma ferramenta que auxilie estudantes da língua portuguesa com baixo custo computacional.

Destaca-se também a ferramenta *mlconjug3*, utilizada para a criação da lista de candidatos do corretor verbal, que se mostrou adequada para tal tarefa. Essa ferramenta conta com uma API de fácil manuseio e um suporte para uma quantidade significativa de línguas, inclusive mostrando uma boa interação com a língua portuguesa.

Já o Jupyter Notebook, por ser uma ferramenta utilizada principalmente por desenvolvedores, dificilmente serviria para um usuário final. Porém, o seu conceito de interação e de células poderia ser reutilizado e adaptado para um futuro trabalho. Além disso, demonstrou-se uma ferramenta bastante eficiente na visualização dos resultados, principalmente em sua interação com a biblioteca SpaCy.

5 CONCLUSÃO

Este trabalho demonstrou a viabilidade da utilização de um sistema híbrido de CEG para correção de erros de concordância verbal, unindo as técnicas baseada em regras e n-grama, a primeira para o reconhecimento de erros e a segunda para a correção dos erros. Além disso, explorou-se o Jupyter Notebook e o SpaCy para a visualização de dados, especialmente através das árvores de dependência.

Por conter uma vasta gama de base de dados disponíveis, o NLTK eficiente para a criação da base de dados utilizadas para a criação do modelo estatístico n-grama. Já as ferramentas de PLN do SpaCy mostraram-se suficientes e úteis para auxiliar em diversas tarefas estudadas e desempenhadas neste trabalho. Essa biblioteca possui uma abrangente *pipeline* e conta com um fácil manuseio.

O método híbrido mostrou-se capaz de realizar tanto a tarefa de reconhecimento de erros, quanto o trabalho de efetuar a correção deles. A base de dados que foi utilizada para a criação do modelo n-grama mostrou-se suficiente para a tarefa de correções de concordância verbal para orações simples.

A abordagem baseada em regras contou com um bom desempenho em reconhecer os sujeitos e os verbos das orações, assim como seus erros. Os resultados obtidos na análise de concordância verbal, que é quando se avalia se o sujeito e o verbo concordam, obteve um *recall* de 94%, uma precisão de 89% e um *f-score* de 91%. Medidas essas que mostram a eficiência da técnica em casos de detecção dos erros, pois, ela alcançou resultados semelhantes ou melhores em comparação com detectores criados para outras línguas.

A tarefa de correção verbal obteve um resultado satisfatório, visto as suas limitações e a sua complexidade em sua avaliação. Para tal tarefa, foram utilizadas duas abordagens diferentes do método n-grama. Essas abordagens contaram com resultados bem próximas, porém, com destaque para a que contou com seus n-grama sendo as classes gramaticais do sujeito e seu verbo *lematizado*.

O problema quanto ao aprendizado e correto uso da concordância verbal é de suma importância para a correta comunicação no português. As tarefas e os estudos desenvolvidos neste trabalho poderão contribuir para o ensino, estudo e correção da concordância verbal no português.

5.1 Trabalhos futuros

Planeja-se a inclusão de outras regras gramáticas não abordadas neste estudo. Entre essas outras regras, estão: casos especiais da concordância verbal, concordância nominal, o uso da

crase e também as regras que diferenciam o uso de certas palavras, como as de: vêm e veem; mau e mal; por que, por quê, porque e porquê.

A partir dos resultados obtidos com esta técnica, de detecção e de correção de erros híbrida, abre-se a oportunidade para a implementação de uma ferramenta, que auxilie tanto professores quanto alunos do português no estudo de concordância verbal. Concomitante a isso, também realizar uma pesquisa de interesse, entendimento e usabilidade do mesmo, tanto para nativos da língua, quanto para aqueles que visam a aprender o português como uma segunda língua.

Apesar das limitações do sistema, especialmente quanto a correção, nota-se que existe a possibilidade da avaliação não ter falhado em marcar alguns casos como corretos. Dito isso, planeja-se realizar uma avaliação dessas correções com juízes humanos.

REFERÊNCIAS

- ABHISHEEK. **A brief introduction to spaCy using python: Production grade NLP library**. 2020. Disponível em: <<https://medium.com/analytics-vidhya/a-brief-introduction-to-spacy-using-python-production-grade-nlp-library-9e3eeb574fa4>>. Acesso em: 07/12/2021. Citado na página 27.
- ALLEN, James F. Natural language processing. In: **Encyclopedia of computer science**. [S.l.: s.n.], 2003. p. 1218–1222. Citado na página 16.
- ALVES, Elder P Maia. A expansão da internet no brasil: Digitalização, mercado e desigualdades sociodigitais the expansion of the internet in brazil: Digitalization, the market and socio-digital inequalities. **Revista Pós Ciências Sociais**, v. 18, n. 2, p. 381–410, 2021. Citado na página 12.
- AZEVEDO, Victor Ribeiro de *et al.* Identificação do perfil de clientes utilizando redes neurais convolucionais. Universidade do Estado do Rio de Janeiro, 2020. Citado na página 24.
- BAIA, Jardel; PRATES, Arley; CLARO, Daniela. Conll dependency parser: Extrinsic evaluation through the open information extraction task. In: SBC. **Anais do VIII Symposium on Knowledge Discovery, Mining and Learning**. [S.l.], 2020. p. 193–200. Citado na página 19.
- BATISTA, Gustavo; DAMASCENO, Adriana. Análise dos motivos para o desuso de recursos computacionais por professores de escolas públicas. In: SBC. **Anais do XXV Workshop de Informática na Escola**. [S.l.], 2019. p. 859–868. Citado na página 12.
- BECHARA, Evanildo. **Moderna gramática portuguesa**. [S.l.]: Nova Fronteira, 2012. Citado na página 15.
- BHIRUD, Nivedita S; BHAVSAR, RP; PAWAR, BV *et al.* A survey of grammar checkers for natural languages. **Computer Science & Information Technology**, p. 51, 2017. Citado na página 43.
- BIDERMAN, Maria Tereza Camargo. Léxico e vocabulário fundamental. **ALFA: Revista de Linguística**, v. 40, 1996. Citado na página 16.
- BOROȘ, Tiberiu; DUMITRESCU, Stefan Daniel; ZAFIU, Adrian; MITITELU, Verginica Barbu; VĂDUVA, Ionut Paul. Racai gec—a hybrid approach to grammatical error correction. In: **Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task**. [S.l.: s.n.], 2014. p. 43–48. Citado 4 vezes nas páginas 13, 20, 21 e 45.
- BRASIL. **Constituição da República Federativa do Brasil: promulgada em 5 de outubro de 1988. 4. ed. São Paulo: Saraiva**. [S.l.: s.n.], 1990. Citado na página 11.
- BRAVO-CANDEL, Daniel; LÓPEZ-HERNÁNDEZ, Jérica; GARCÍA-DÍAZ, José Antonio; MOLINA-MOLINA, Fernando; GARCÍA-SÁNCHEZ, Francisco. Automatic correction of real-word errors in spanish clinical texts. **Sensors**, MDPI, v. 21, n. 9, p. 2893, 2021. Citado 2 vezes nas páginas 45 e 46.
- BRUNIALTI, Lucas; PERES, Sarajane; FREIRE, Valdinei; LIMA, Clodoaldo. Aprendizado de maquina em sistemas de recomendacao baseados em conteudo textual: Uma revisao sistematica. **Anais do XI Simpósio Brasileiro de Sistemas de Informação**, SBC, p. 203–210, 2015. Citado na página 22.

CORREA, Weliton. **Frases PT-BR**. 2022. Disponível em: <<https://www.kaggle.com/datasets/weliton2/frases-ptbr>>. Acesso em: 17/06/2022. Citado na página 41.

COSTA, Pablo Botton da. Análise sintática semi-supervisionada por constituição aplicada ao português e inglês. Universidade Federal do Pampa, 2014. Citado na página 18.

DIAO, Sekou. mlconjug3. **GitHub**. Note: <https://github.com/SekouDiaoNlp/mlconjug3> Cited by, 2021. Citado 3 vezes nas páginas 27, 33 e 34.

DORASH, Maryna. **Machine learning vs. rule based systems in NLP**. Friendly Data, 2017. Disponível em: <<https://medium.com/friendly-data/machine-learning-vs-rule-based-systems-in-nlp-5476de53c3b8>>. Citado na página 20.

EXPLOSION. **Trained Models & Pipelines - Spacy Models Documentation**. 2021. Disponível em: <<https://spacy.io/models>>. Acesso em: 28/02/2022. Citado na página 26.

FAHDA, Asanilta; PURWARIANTI, Ayu. A statistical and rule-based spelling and grammar checker for indonesian text. In: IEEE. **2017 International Conference on Data and Software Engineering (ICoDSE)**. [S.l.], 2017. p. 1–6. Citado na página 13.

FARINON, João Luís. Análise e classificação de conteúdo textual. 2015. Citado 3 vezes nas páginas 15, 16 e 20.

FERRÃO, Marco F; MELLO, Cesar; BORIN, Alessandra; MARETTO, Danilo A; POPPI, Ronei J. Ls-svm: uma nova ferramenta quimiométrica para regressão multivariada. comparação de modelos de regressão ls-svm e pls na quantificação de adulterantes em leite em pó empregando nir. **Química Nova**, SciELO Brasil, v. 30, n. 4, p. 852–859, 2007. Citado na página 23.

FERREIRA, Hugo Honda. Processamento de linguagem natural e classificação de textos em sistemas modulares. 2019. Citado na página 43.

FILHO, José Lopes Moreira. **Linguística e computação em diálogo para análise de textos e criação de atividades de leitura em língua inglesa**. Tese (Doutorado) — Universidade de São Paulo, 2015. Citado na página 29.

FONSECA, Erick; ROSA, João Luís G. Mac-morpho revisited: Towards robust part-of-speech tagging. In: **Proceedings of the 9th Brazilian symposium in information and human language technology**. [S.l.: s.n.], 2013. Citado na página 28.

FONSECA, Erick R; ROSA, João Luís G; ALUÍSIO, Sandra Maria. Evaluating word embeddings and a revised corpus for part-of-speech tagging in portuguese. **Journal of the Brazilian Computer Society**, Springer, v. 21, n. 1, p. 1–14, 2015. Citado na página 35.

FORNAZARI, João Eduardo Kozan Silva. **RALeM: uma ferramenta para auxiliar na identificação de palavras significativas em conversas em ferramentas on-line colaborativas através de análise léxico-morfológica**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2021. Citado na página 17.

GEEKS, Geeks for. **Stochastic Gradient Descent (SGD)**. 2021. Disponível em: <<https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>>. Acesso em: 08/06/2022. Citado na página 23.

GEEKS, Geeks for. **Defaultdict in Python**. 2022. Disponível em: <<https://www.geeksforgeeks.org/defaultdict-in-python/?ref=lbp>>. Acesso em: 04/06/2022. Citado na página 36.

GONÇALVES, André Ricardo. **Máquina de vetores suporte**. 2010. Disponível em: <<https://andreric.github.io/files/pdfs/svm.pdf>>. Acesso em: 01/05/2022. Citado na página 23.

GONZALEZ, Marco; LIMA, Vera Lúcia Strube. Recuperação de informação e processamento da linguagem natural. In: **XXIII Congresso da Sociedade Brasileira de Computação**. [S.l.: s.n.], 2003. v. 3, p. 347–395. Citado 2 vezes nas páginas 16 e 18.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Data Science Academy, 2016. Disponível em: <<https://www.deeplearningbook.com.br/aprendizado-com-a-descida-do-gradiente>>. Acesso em: 17/05/2022. Citado 2 vezes nas páginas 23 e 24.

HEREDIA, Érica. **Classes Gramaticais**. Quero Educação, 2018. Disponível em: <<https://querobolsa.com.br/enem/portugues/classes-gramaticais>>. Acesso em: 01/02/2022. Citado na página 17.

HONNIBAL, Matthew; MONTANI, Ines; LANDEGHEM, Sofie Van; BOYD, Adriane. **spaCy: Industrial-strength Natural Language Processing in Python**. Zenodo, 2020. Disponível em: <<https://doi.org/10.5281/zenodo.1212303>>. Citado na página 26.

HOSCH, William L. **Machine Learning**. 2021. Disponível em: <<https://www.britannica.com/technology/machine-learning>>. Acesso em: 08/06/2022. Citado na página 22.

HUANG, Guimin; FAN, Chunli; SUN, Zhenglin. A hybrid correction method of english grammar error for chinese learners' essays. In: **Information Technology and Intelligent Transportation Systems**. [S.l.]: IOS Press, 2019. p. 1–10. Citado na página 13.

HUNTERMANN, Caroline; CUNHA, Karina Zendron Da. O ensino de concordância verbal no ensino médio: uma análise em livros didáticos e em uma plataforma online de língua portuguesa. **Miguilim-Revista Eletrônica do Netlli**, v. 10, n. 1, p. 85–106, 2021. Citado na página 15.

IGISCK, Nadine; SILVA, Guilherme Nunes da; RODRIGUES, Andressa Xavier; SCHUMACHER, Jane; ERICHSEN, Ronaldo; GARNERO, Analía. O uso de tecnologias de informação e comunicação (tics) na educação especial. **Anais do Salão Internacional de Ensino, Pesquisa e Extensão**, v. 9, n. 3, 2017. Citado na página 13.

KANG, Ning; SINGH, Bharat; AFZAL, Zubair; MULLIGEN, Erik M van; KORS, Jan A. Using rule-based natural language processing to improve disease normalization in biomedical text. **Journal of the American Medical Informatics Association**, BMJ Publishing Group, v. 20, n. 5, p. 876–881, 2013. Citado na página 20.

KAPADIA, Shashank. **Language models: N-gram**. Towards Data Science, 2019. Disponível em: <<https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9>>. Acesso em: 01/04/2022. Citado na página 21.

KENOBI, Obi-Wan. **Star Wars — Wikiquote, a coletânea de citações livre**. 1977. Disponível em: <[//pt.wikiquote.org/w/index.php?title=Star_Wars&oldid=158357](https://pt.wikiquote.org/w/index.php?title=Star_Wars&oldid=158357)>. Acesso em: 08/06/2022. Citado na página 16.

- KOSTAREVA, Taisiya; CHUPRINA, Svetlana; NAM, Alexandr. Using ontology-driven methods to develop frameworks for tackling nlp problems. In: **AIST (Supplement)**. [S.l.: s.n.], 2016. p. 102–113. Citado na página 16.
- KRESS, Gunther. **Literacy in the new media age**. [S.l.]: Routledge, 2003. Citado na página 12.
- LEAL, Sidney Evaldo; MAGALHAES, Vanessa Maia Aguiar de; DURAN, Magali Sanches; ALUÍSIO, SM. Avaliação automática da complexidade de sentenças do português brasileiro para o domínio rural. In: IN: SYMPOSIUM IN INFORMATION AND HUMAN LANGUAGE TECHNOLOGY AND COLLOCATES [S.l.], 2019. Citado na página 33.
- LIN, Nankai; LIN, Xiaotian; YANG, Ziyu; JIANG, Shengyi. A new evaluation method: Evaluation data and metrics for chinese grammar error correction. **arXiv preprint arXiv:2205.00217**, 2022. Citado na página 46.
- LOPER, Edward; BIRD, Steven. Nltk: The natural language toolkit. **arXiv preprint cs/0205028**, 2002. Citado na página 27.
- LORENA, Ana Carolina; CARVALHO, André CPLF De. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007. Citado na página 23.
- LUCCA, JL De; NUNES, Maria das Graças Volpe. Lematização versus stemming. **USP, UFSCar, UNESP, São Carlos, São Paulo**, 2002. Citado na página 33.
- LUFT, Celso Pedro. **Moderna gramática brasileira**. [S.l.]: Globo Livros, 2002. 27 p. Citado na página 17.
- LUNA-RAMÍREZ, Wulfrano A; JAIMEZ-GONZÁLEZ, Carlos R. A proposal of automatic error correction in text. **arXiv preprint arXiv:2112.01846**, 2021. Citado 2 vezes nas páginas 13 e 45.
- LYONS, John. **Natural Language and Universal Grammar: Volume 1: Essays in Linguistic Theory**. [S.l.]: Cambridge University Press, 1991. v. 1. Citado na página 15.
- MATTELART, Armand. **História das teorias da comunicação**. [S.l.]: Edições Loyola, 2011. Citado na página 11.
- MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, Manole, v. 1, n. 1, p. 32, 2003. Citado na página 22.
- NAPOLIS, Courtney; SAKAGUCHI, Keisuke; POST, Matt; TETREAULT, Joel. Ground truth for grammatical error correction metrics. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**. [S.l.: s.n.], 2015. p. 588–593. Citado na página 46.
- NASCIMENTO, Emerson Ribeiro da Silva do *et al.* 11. o pretérito mais-que-perfeito como ferramenta de inclusão da gramática histórica em sala de aula. **Revista Philologus**, v. 27, n. 80, p. 173–83, 2021. Citado na página 34.

- NHATUVE, Diocleciano; CHIPARA, Margret. Aspectos de concordância verbal na aprendizagem do português língua estrangeira. **Mandinga-Revista de Estudos Linguísticos**, v. 1, n. 2, p. 8–14, 2017. Citado na página 12.
- NUNES, Daniel Vitor de Oliveira. A importância da leitura para o desenvolvimento cognitivo: o papel do bibliotecário escolar no desenvolvimento do ser. Universidade Federal da Paraíba, 2018. Citado na página 11.
- ONU. **ONU no Brasil | Mundo terá mais de 500 milhões de falantes do português até final do século, diz CPLP**. United Nations, 2021. Disponível em: <<https://brasil.un.org/pt-br/126262-mundo-tera-mais-de-500-milhoes-de-falantes-do-portugues-ate-final-do-seculo-diz-cplp>>. Acesso em: 20/04/2022. Citado na página 15.
- PADOVANI, Djalma. **Um método adaptativo para análise sintática do Português Brasileiro**. Tese (Doutorado) — Universidade de São Paulo, 2022. Citado na página 12.
- PERLINA, AB; TSYGANKOVA, YN. How to learn a foreign language in the internet. Polesky State University, 2017. Citado na página 12.
- PHILLIPS, Tom. **Brazil: Bolsonaro’s education Minister ridiculed for series of spelling HOWLERS**. Guardian News and Media, 2020. Disponível em: <<https://www.theguardian.com/world/2020/jan/09/brazil-bolsonaro-spelling-education-minister-abraham-weintraub>>. Acesso em: 10/03/2022. Citado na página 12.
- PIMENTEL, João Felipe; OLIVEIRA, Gabriel P; SILVA, Mariana O; SEUFITELLI, Danilo B; MORO, Mirella M. Ciência de dados com reprodutibilidade usando jupyter. **Sociedade Brasileira de Computação**, 2021. Citado na página 29.
- PINHEIRO, Pedro; AMARIS, Marcos. Classificação dos códigos de ncm usando processamento de linguagem natural. In: SBC. **Anais da I Escola Regional de Alto Desempenho Norte 2 e I Escola Regional de Aprendizado de Máquina e Inteligência Artificial Norte 2**. [S.l.], 2021. p. 9–12. Citado na página 43.
- PINTO, Sara Catarina Silva. **Processamento de linguagem natural e extração de conhecimento**. Tese (Doutorado) — Universidade de Coimbra, 2015. Citado na página 15.
- PUTRA, FP; ENDA, D. Model design for grammatical error identification in software requirements specification using statistics and rule-based techniques. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [S.l.], 2020. v. 1450, n. 1, p. 012071. Citado 3 vezes nas páginas 13, 21 e 46.
- RADEMAKER, Alexandre; CHALUB, Fabricio; REAL, Livy; FREITAS, Cláudia; BICK, Eckhard; PAIVA, Valeria de. Universal dependencies for portuguese. In: **Proceedings of the Fourth International Conference on Dependency Linguistics (Depling)**. Pisa, Italy: [s.n.], 2017. p. 197–206. Disponível em: <<http://aclweb.org/anthology/W17-6523>>. Citado na página 26.
- REMIGIO, Matheus. Máquina de vetores de suporte — svm. **Medium**, 2020. Disponível em: <<https://medium.com/@msremigio/máquinas-de-vetores-de-suporte-svm-77bb114d02fc>>. Acesso em: 08/06/2022. Citado 2 vezes nas páginas 23 e 24.

RICH, Elaine; KNIGHT, Kevin; CALERO, Pedro Antonio González; BODEGA, Fernando Trescastro. **Inteligência artificial**. [S.l.]: McGraw-Hill, 1994. v. 1. Citado na página 18.

ROQUE, Carolinne; JÚNIOR, Maurilio Martins Campano; BARBOSA, Cinthyan Renata Sachs C de *et al.* Sistema de apoio à decisão por pln para consultas de pragas na cultura da soja. In: SBC. **Anais do XLVI Seminário Integrado de Software e Hardware**. [S.l.], 2019. p. 45–56. Citado na página 16.

ROZOVSKAYA, Alla; ROTH, Dan. Grammar error correction in morphologically rich languages: The case of russian. **Transactions of the Association for Computational Linguistics**, MIT Press, v. 7, p. 1, 2019. Citado na página 12.

SAKURAI, GUILHERME YUKIO. Processamento de linguagem natural-detecção de fake news. 2019. Citado 2 vezes nas páginas 18 e 30.

SANTANA, Raíssa Silva. **Análise sintática**. 2020. Disponível em: <<https://www.infoescola.com/portugues/analise-sintatica/>>. Acesso em: 04/06/2022. Citado na página 17.

SARRAZIN, Pierre. **Verbiste**. 2020. Disponível em: <<http://perso.b2b2c.ca/~sarrazip/dev/verbiste.html>>. Acesso em: 08/06/2022. Citado na página 27.

SCIKITLEARN. **ElasticNet**. 2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html>. Acesso em: 08/06/2022. Citado na página 33.

SCIKITLEARN. **Stochastic Gradient Descent**. 2022. Disponível em: <<https://scikit-learn.org/stable/modules/sgd.html>>. Acesso em: 08/06/2022. Citado na página 24.

SHAALAN, Khaled. Rule-based approach in arabic natural language processing. **The International Journal on Information and Communication Technologies (IJICT)**, v. 3, n. 3, p. 11–19, 2010. Citado na página 20.

ŠOIĆ, Renato; VUKOVIĆ, Marin; JEŽIĆ, Gordan. Spoken notifications in smart environments using croatian language. **Computer Science and Information Systems**, v. 18, n. 1, p. 231–250, 2021. Citado na página 22.

SOUZA, Karina. **A Cada Segundo, 14 Pessoas Começam a USAR Uma Rede Social Pela 1ª vez**. 2020. Disponível em: <<https://exame.com/tecnologia/a-cada-segundo-14-pessoas-comecam-a-usar-uma-rede-social-pela-1a-vez/>>. Acesso em: 10/09/2021. Citado na página 12.

SOUZA, Nahim Alves de. Aumentando o poder preditivo de classificadores lineares através de particionamento por classe. Universidade Federal de São Carlos, 2018. Citado 2 vezes nas páginas 22 e 23.

SRINIVASAN, Aishwarya V. **Stochastic Gradient Descent — Clearly Explained**. 2019. Disponível em: <<https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31>>. Acesso em: 08/06/2022. Citado 2 vezes nas páginas 24 e 25.

STOLCKE, Andreas; SEGAL, Jonathan. Precise n-gram probabilities from stochastic context-free grammars. **arXiv preprint cmp-lg/9405016**, 1994. Citado na página 21.

- TADEU, Vinícius. **Miguel e Helena são os nomes mais registrados no Brasil em 2021**. 2021. Disponível em: <<https://www.cnnbrasil.com.br/nacional/miguel-e-helena-sao-os-nomes-mais-registrados-no-brasil-em-2021-veja-lista/>>. Acesso em: 01/04/2022. Citado na página 41.
- VAPNICK, Vladimir N. **Statistical learning theory**. [S.l.]: Wiley, New York, 1998. Citado na página 22.
- VARGAS, Natalie Lourenço; KEPLER, Fabio Natanael. Análise sintática automática por dependência. **Anais do Salão Internacional de Ensino, Pesquisa e Extensão**, v. 4, n. 2, 2012. Citado na página 18.
- VASCONCELLOS-SILVA, Paulo; ARAUJO-JORGE, Tania. Análise de conteúdo por meio de nuvem de palavras de postagens em comunidades virtuais: novas perspectivas e resultados preliminares. **CIAIQ2019**, v. 2, p. 41–48, 2019. Citado na página 41.
- VRECH, Giovani Müller; SAITO, Rodrigo Kiyoshi; CÂMARA, Carlos Eduardo. Processamento de linguagem natural na resolução de problemas de classificação. **Revista de Ubiquidade**, v. 4, n. 1, p. 54–93, 2021. Citado na página 43.
- W3SCHOOLS. **Python Dictionaries**. 2022. Disponível em: <https://www.w3schools.com/python/python_dictionaries.asp>. Acesso em: 04/04/2022. Citado na página 36.
- WANG, Yu; WANG, Yuelin; LIU, Jie; LIU, Zhuo. A comprehensive survey of grammar error correction. **arXiv preprint arXiv:2005.06600**, 2020. Citado na página 20.
- WEBSTER, Jonathan J; KIT, Chunyu. Tokenization as the initial phase in nlp. In: **COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics**. [S.l.: s.n.], 1992. Citado na página 16.
- WIECHETEK, Linda; PIRINEN, Flammie; HÄMÄLÄINEN, Mika; ARGESE, Chiara *et al.* Rules ruling neural networks—neural vs. rule-based grammar checking for a low resource language. In: INCOMA. **Proceedings of the International Conference Recent Advances In Natural Language Processing 2021**. [S.l.], 2021. Citado 2 vezes nas páginas 13 e 43.
- XIANG, Yang; YUAN, Bo; ZHANG, Yaoyun; WANG, Xiaolong; ZHENG, Wen; WEI, Chongqiang. A hybrid model for grammatical error correction. In: **Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task**. [S.l.: s.n.], 2013. p. 115–122. Citado na página 20.
- YADAV, Ajay. Support vector machines (svm). **Towards Datas Science**, 2018. Disponível em: <<https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589>>. Acesso em: 08/06/2022. Citado na página 23.
- YEH, Jui-Feng; CHANG, Li-Ting; LIU, Chan-Yi; HSU, Tsung-Wei. Chinese spelling check based on n-gram and string matching algorithm. In: **Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)**. [S.l.: s.n.], 2017. p. 35–38. Citado na página 13.